CENTER FOR
MACHINE PERCEPTION

CZECH TECHNICAL
UNIVERSITY

Nonlinear Diffusion for Image Filtering and Monotonicity Enhancement

Pavel Mrázek

mrazekp@cmp.felk.cvut.cz

CTU–CMP–2001–19

25/6/2001

Available at
ftp://cmp.felk.cvut.cz/pub/cmp/articles/mrazek/Mrazek-phd01.pdf

Thesis Advisor: Mirko Navara

This research was supported by the Czech Ministry of Education
under Project LN00B096.

Research Reports of CMP, Czech Technical University in Prague, No. 19, 2001

Published by

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

PhD THESIS

ISSN 1213-2365

Nonlinear Diffusion for Image Filtering and Monotonicity Enhancement

PhD Thesis

Pavel Mrázek

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Praha 6, Czech Republic

www: `http://cmp.felk.cvut.cz/˜mrazekp`
e-mail: `mrazekp@cmp.felk.cvut.cz`

To Alča

Preface

Malé poměry nevznikají tím, že máme malá města, nýbrž tím, že
jsme v nich neradi. Hledá-li člověk romantické dobrodružství,
je mu v takovém případě jeho zahrádka ovšem příliš malá;
ale chce-li uhrabat cestičky nebo vyplejt bejlí, shledá,
že je zatraceně veliká, větší, než vůbec tušil.[1]

Karel Čapek: O malých poměrech.
Přítomnost 1925/1926, č. 48, 10. 12. 1925.

This is a PhD thesis about filtering of images and other similar kinds of data. Before starting to read or browse the pages, the reader is likely to ask one important question: after so much effort spent on this problem in the last ten, twenty, forty years, what is the reason for writing even a single more page? What contribution does it make?

Postponing precise contribution statements for a moment, we have to warn you at the very beginning: if you hope to witness a breakthrough, an opening of a brand new theory, or some other kind of 'romantic adventure', you will not find it in this work. Rather, in the sense of Čapek's words quoted above, we entered a garden which other people had built and developed; we tried to smooth the paths, rake over the soil and weed some flower beds. The garden was big enough, and being busy with these tasks, we did not mind we were not involved in the discovery of new continents; hopefully, our time and effort has helped to cultivate one or two small and useful plants.

The work would be impossible, or at least much less enjoyable to complete without the support and assistance of many people, either colleagues, friends, or relatives. The author is grateful to all of them. From those involved in the scientific part of the project, I owe much to Mirko Navara, my thesis supervisor at the Center for Machine Perception, Czech Technical University: he supported me during all those years, from directing my attention towards the topics of this thesis at the

[1] "Provincialism does not arise from our living in small towns, but from our dislike of living there. If a man seeks romantic adventure, his garden obviously appears too small to him; but wanting to rake the paths or pull out weeds, he will find the garden damn big, bigger than he could ever imagine." Excerpt from a magazine article by Karel Čapek, Czech playwright, journalist and writer (1890–1938).

beginning, to expressing many helpful comments on the manuscript. I would like to acknowledge Karol Mikula from the Slovak University of Technology who was the first to introduce nonlinear diffusion concept to me; I drew much inspiration and encouragement from the discussions with him and other people involved in diffusion filtering whom I have had the chance to meet: Joachim Weickert, Eberhard Bänsch, Pietro Perona, Alessandro Sarti and Zuzka Krivá are just a few of them. Many thanks must also go to all my colleagues at the Center for Machine Perception who helped to create the friendly and motivating atmosphere I have had the pleasure to experience.

Prague, June 2001 Pavel Mrázek

A technical note: If you are reading the thesis in its electronic form as a PDF document, the references to chapters, equations, citations etc. are for your convenience linked to the referenced destination. If your viewer supports this feature (and the Adobe Acrobat Reader does), you can follow links and references easily by a mouse click, and come back using the viewer's navigation.

The document is available at
`ftp://cmp.felk.cvut.cz/pub/cmp/articles/mrazek/Mrazek-phd01.pdf`
or through my homepage `http://cmp.felk.cvut.cz/~mrazekp`.

Contents

1 Introduction

1.1 Motivation

Reconstruction of full three-dimensional shape of real objects from visual information, i.e. from essentially two-dimensional data, belongs to main domains of computer vision applications. Only a brief introduction of all the methods which have been employed would result in a long text, we will mention here only a few (possibly the most) important groups.

The main representative of passive vision, *stereo*, starts from (at least) two images of a scene, taken from different positions (either by two or more cameras, or by one moving camera). Then from the known change of camera positions between the two images and by finding the corresponding points (belonging to the same object/feature) in both images it is possible to compute angles from cameras to the point in 3D space and thus reconstruct its 3D position.

Stereo is intrinsically a difficult task, partly because of the computationally expensive search for corresponding points in different images. Where possible, it is helpful to restrict the generality of the approach and try a simpler one. So *active vision* techniques came into play for moderately sized scenes; with them, the correspondence problem is avoided by introducing special light conditions artificially. Various structured light types can be used. Perhaps the most commonly used *laser-plane range finder* lights the observed scene by a single plane of laser light, a single strip of light is then seen in the camera. From the known mutual position of laser and camera and the position of bright pixels in the image two angles are obtained and the 3D shape of the scene is reconstructed.

The major advantage of this approach lies in its simplicity and accuracy (see [26]); on the other hand it reveals a drawback that a single plane is illuminated by the laser light at one moment, and only points residing in that plane can be reconstructed from a single image. Many images (which implies longer time) are needed to capture a complete scene, which makes the method unusable for any moving, non-rigid or non-stationary objects.

A different lighting scheme may provide a better solution. In [9] several possi-

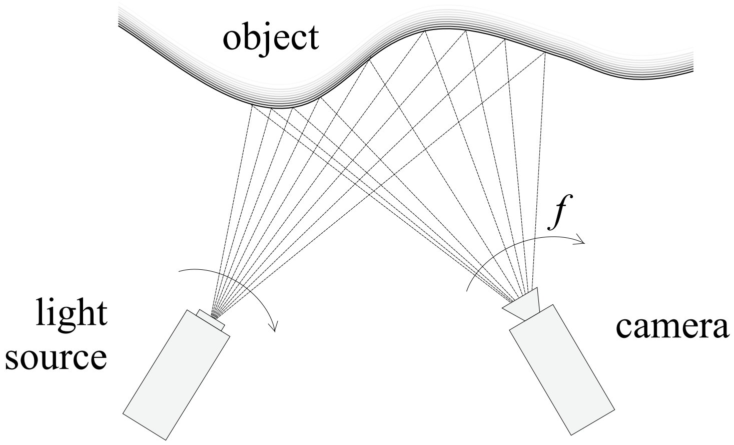


Figure 1.1: Rainbow range finder - a system for 3D reconstruction by active vision: light of monotonically increasing wavelength is projected into the scene and full 3D shape is reconstructed from the colour image observed by the camera.

bilities are reviewed and a colour light coding method (using projection of several planes) developed as a compromise between accuracy of the reconstruction and the acquisition period.

If we want to accomplish the reconstruction from a single image, we may employ colour coded illumination in a system called *rainbow range finder* [39, 35, 36]: the light projected into the scene forms a rainbow with the wavelength increasing in one direction (see Figure 1.1 for a sketch of the setting).

As the colours are created by decomposing white light by an interference filter, each plane consists of light of a single wavelength; then the colour in the image observed by the camera is not affected by reflections from the scene and should also exhibit the property of monotonically increasing wavelength in one direction, at least piecewise, with discontinuities at object boundaries. We should be able to assign a point in the image to a single plane passing through the light projector uniquely, based only on the observed colour of that point. The plane's angle from the projector, α, together with the point's location in the image, corresponding to angle β, and with the relative position of the projector and the camera determine the 3D, real-world position of that point. However, the unique and precise identification of the point's position from its observed colour is complicated by an imprecision in colour measurement and by other kinds of noise present in the experiment. The data which should be piecewise continuous and piecewise increasing in one direction do not always comply with this rule.

This situation (piecewise continuous and piecewise increasing data corrupted by noise) is not restricted to the rainbow range finder discussed so far: similar properties are valid for other types of range finders (using laser-plane, structured light, etc.) as well. And, more important, similar properties can be also assumed for other types of data: the depths of some objects in the scene from a measurement device increase or decrease continuously on continuous surfaces and change abruptly at object boundaries, grey values in some types of images change gradually in homogeneous regions and suddenly across edges, etc.

1.2 Goal of the thesis

In this text we assume that there exists an ideal function $\widetilde{\mathbf{f}}$ which is piecewise continuous and piecewise monotone (i.e. piecewise increasing or decreasing; see Chapter 4 for a more detailed definition). This ideal information is observed by an imperfect measurement device and we obtain discrete and noisy data $\mathbf{f} = (f_{i,j})$; the acquisition process is modeled by

$$f_{i,j} = h_{i,j} * \widetilde{\mathbf{f}} + n, \qquad x_i = i \cdot \Delta x, \;\; y_j = j \cdot \Delta y \tag{1.1}$$

where $h_{i,j}$ is the sampling kernel for position x_i, y_j, and $\Delta x, \Delta y$ are the sampling intervals in the directions of axes x and y, respectively. Some noise n is added to the samples during the discretization.

The ideal goal of our work is to replace the noisy measurements $\mathbf{f}$ by some other values $\widehat{\mathbf{f}}$ which would contain all the useful information of $\widetilde{\mathbf{f}}$ but discard the noise. Unfortunately, it is generally not possible to accomplish the task. Mathematically speaking, the problem is ill posed[1]: we cannot distinguish the contribution of the noise n from the contribution of the signal in $\mathbf{f}$. Also, the information about the positions of the discontinuities in $\widetilde{\mathbf{f}}$ is irreversibly lost by the discretization.

To achieve a more realistic task, we have to add and employ some more information, in other words we have to regularize the problem. So we will try to *filter* the data $\mathbf{f}$, i.e. to smooth and simplify the measurements, hoping that if we use a good filtering procedure which reflects the assumptions about the desired data properties (as mentioned above), we will be able to remove the noise and get close to the original function $\widetilde{\mathbf{f}}$. The regularization, or the added information will consist in our case in choosing the method for the filtering, in adapting the method for the data properties, and in selecting the parameters of the method. To be more specific, the **main goal of the thesis is to develop a filtering procedure suitable to remove noise from data which should be piecewise continuous and piecewise monotone while preserving (or enhancing) the desired data properties**.

1.3 Thesis contributions

This thesis is mainly about nonlinear diffusion, a powerful filtering technique, created at an intersection of many strong fields such as scale space, energy minimization, modeling of physical phenomena, robust statistics, and mathematical morphology. Consistently with the stated goal, the **main contribution of the thesis**

[1] In the sense of Hadamard, a problem is well posed if there exists a unique solution and if the solution depends continuously on the measurements. Otherwise, the problem is ill posed [41, 5].

is the design of a monotonicity-enhancing nonlinear diffusion filter. The classical NL diffusion filters are described by partial differential equations which use piecewise constant functions as their model. We extend the filter to piecewise monotone (or, ultimately, piecewise linear) data by considering directional derivatives of the original function. We discuss the ideas, develop the algorithms, and analyse the theoretical properties of the new method in Chapter 4.

On the pursuit of this final goal, we learned that not all of the questions linked to the nonlinear diffusion filtering had been answered. So in Chapter 3 we offer a novel approach to **selection of optimal stopping time** for NL diffusion, based on the claim that it is reasonable to minimize the correlation between signal and noise resulting from the filtering procedure; see Section 3.1 for details. In the same chapter, we also analyse the possibilities of **consistent positive directional splitting** of anisotropic NL diffusion into four directions, and summarize various ideas into an **autonomous diffusion filtering algorithm**.

1.4 Overview of the chapters

The structure of the thesis is as follows:

- Chapter 2 introduces nonlinear diffusion and summarizes previous work in this quickly developing field; it classifies the methods, discusses the role of the parameters, and offers a brief overview of numerical methods for NL diffusion.

- Chapter 3 offers several smaller contributions the author has made to NL diffusion; perhaps the most important is the choice of the diffusion stopping time, other topics include directional splitting of anisotropic diffusion on a 3×3 window, and a summary of an autonomous diffusion filtering algorithm.

- Chapter 4 develops the monotonicity-enhancing nonlinear diffusion filter as motivated above. The main idea is to filter the directional derivatives instead of the original data; such an approach removes noise while preserving both discontinuities and trends of growth present in the data.

- Some experimental results are shown in Chapter 5, both for the classical and the monotonicity-enhancing nonlinear diffusion procedures.

- Finally, the conclusions and possible directions for further research are proposed in Chapter 6.

2 Nonlinear diffusion filtering (state of the art)

2.1 Introduction

In this chapter we introduce nonlinear diffusion, a powerful image filtering method, particularly suitable for piecewise constant data and additive noise. This introduction should also survey the "state of the art", i.e. to give a brief overview of the field before we entered it, in order to anchor our contributions to (and distinguish them from) previous work of other authors.

Let $\widetilde{\mathbf{f}}\colon \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ be an unknown real function, in the following we will assume that $\widetilde{\mathbf{f}}$ is piecewise continuous and reasonably smooth. More knowledge on the function properties would be highly useful: piecewise constancy and piecewise monotonicity will be explored, discussed and exploited later in this text.

The function $\widetilde{\mathbf{f}}$ is sampled and represented by a 2D array of values $\mathbf{f} = (f_{i,j})$,[1] our input data:

$$f_{i,j} = h_{i,j} * \widetilde{\mathbf{f}} + n, \qquad x_i = i \cdot \Delta x, \;\; y_j = j \cdot \Delta y \tag{2.1}$$

where $h_{i,j}$ is the sampling kernel for position x_i, y_j, and $\Delta x, \Delta y$ the sampling intervals in the directions of axes x and y, respectively. Some noise n is added to the samples during the discretization process.

We want to find another set of sampled values, $\mathbf{u} = (u_{i,j})$, such that it contains as much information on the original, noise-free $\widetilde{\mathbf{f}}$ as possible: try to remove the noise but preserve the useful information in $f_{i,j}$. To this end, we first introduce the scale concept both in linear and nonlinear settings, and concentrate in more detail on nonlinear diffusion.

2.2 Linear scale space

The tasks of approximation and noise filtering are closely linked to that of replacing the input by its smoother version with fewer local extrema, and this smoothing or

[1] We will use the bold symbol $\mathbf{f}$ both for a function and for the vector of its discrete samples. The distinction should be clear from the context.

simplification of a signal leads directly to the notion of *scale space*.

Typically, an image contains useful information at various levels of detail, at various object sizes, or at different scales. If we embed the original data, $\mathbf{f}(x)$, $x \in \mathbb{R}^N$, into a family $\{T_t\mathbf{f} \mid t \geq 0\}$ of gradually simplified versions of $\mathbf{f}$, we obtain a *scale space representation* which describes the original signal at a continuum of scales. By the analysis of such a representation at different levels, we may obtain some information which was only implicit in the original data.

The scale space representation is usually required to be endowed with some attributes which can be classified as architectural, stability, and invariance properties [3].

The architectural requirements include the semi-group property, or recursivity

$$\begin{aligned} T_0\mathbf{f} &= \mathbf{f}, \\ T_{t+s}\mathbf{f} &= T_t(T_s\mathbf{f}) \qquad \forall s,t \geq 0, \end{aligned} \tag{2.2}$$

regularity (continuity) of T_t, and locality: for a small t, the value of $T_t\mathbf{f}$ at any point x is determined by the behaviour of $\mathbf{f}$ near x.

The stability property is stated differently by different authors: as a *comparison principle* [3]

$$T_t(\mathbf{f}) \leq T_t(\mathbf{g}) \text{ on } \mathbb{R}^N \qquad \forall t \geq 0, \forall \mathbf{f} \leq \mathbf{g}, \tag{2.3}$$

as a maximum-minimum alias *extremum principle* [45]

$$\inf_{\mathbb{R}^N} \mathbf{f} \leq T_t\mathbf{f} \leq \sup_{\mathbb{R}^N} \mathbf{f}, \tag{2.4}$$

as a causality requirement, nonenhancement of local extrema, or in another way. Generally, these formulations all express the fact that the operator T_t should be information-reducing, smoothing, and should not create any spurious details which were not present in the original data.

The invariance properties formalize the intuitive request that the scale space creation should ideally not depend on the shift in grey values of the input data, on the translation of the original function, etc.

In their work, Alvarez, Guichard, Lions and Morel [3] show that any scale space satisfying some of these natural architectural, stability, and invariance assumptions is formed by a sequence of images $\mathbf{u}(x,t) = (T_t\mathbf{f})(x)$ found by solving the partial differential equation of second order

$$\frac{\partial \mathbf{u}}{\partial t} = F(\nabla^2\mathbf{u}, \nabla\mathbf{u}, t) \tag{2.5}$$

where ∇ is the gradient operator.[2]

[2] Let us introduce some notation here. For data $\mathbf{u} \in \mathbb{R}^N$, the gradient operator ∇ is a column vector of N elements

$$\nabla \equiv \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_N}\right)^T$$

where x_i, $i = 1,\ldots,N$ are the coordinate axes. For the case of normal images, $N = 2$ and the

If we add the *linearity principle*

$$T_t(a\mathbf{f} + b\mathbf{g}) = aT_t\mathbf{f} + bT_t\mathbf{g} \qquad \forall t \geq 0,\ \forall a, b \in \mathbb{R} \tag{2.6}$$

to the set of requirements on T_t, the scale space representation is restricted to a single possibility, and the best explored one: the *linear scale space.*

As pointed out recently by Weickert *et al.* in [51], the 1D linear scale space was derived axiomatically by Iijima as early as in 1959. However, his papers (most of which were written in Japanese) passed unnoticed by the western scientific community, so the Witkin's 1983 paper [56] is usually cited as the beginning of the linear scale space idea. The formalism has been further developed by Koenderink [15]. A comprehensive presentation of the linear scale space can be found e.g. in the collection [40], or in the monograph [17] by Lindeberg, who also extends the linear scale space to discrete signals.

Starting from the data $\mathbf{f}(x)$, $x \in \mathbb{R}^N$, let us add a *scale parameter* t and construct the *scale-space representation* L as follows:

$$\begin{aligned} L(x,0) &= \mathbf{f}(x), \\ L(x,t) &= (G_\sigma * \mathbf{f})(x) = \int_{\mathbb{R}^N} G_\sigma(x-y)\,\mathbf{f}(y)\,\mathrm{d}y. \end{aligned} \tag{2.7}$$

For $t = 0$ the representation is identical with the original data, a higher level $L(x,t)$ is obtained as a convolution of $\mathbf{f}$ and a smoothing kernel G_σ.

The smoothing kernel G_σ is the Gaussian kernel given in the continuous case and arbitrary dimension N by

$$G_\sigma(x) = \frac{1}{(2\pi\sigma^2)^{N/2}}\; e^{-x^T x/2\sigma^2}. \tag{2.8}$$

The width (standard deviation) σ of the Gaussian is linked to the 'time' of the linear scale space L by the expression $\sigma = \sqrt{2t}$.

There is also an alternative way to construct the linear scale space: for any bounded function $\mathbf{f}$, the linear diffusion process, or *heat equation*

$$\partial_t \mathbf{u} = \Delta \mathbf{u} \tag{2.9}$$

$$\mathbf{u}(x,0) = \mathbf{f}(x) \tag{2.10}$$

image gradient is

$$\nabla \mathbf{u} = \left(\frac{\partial \mathbf{u}}{\partial x}, \frac{\partial \mathbf{u}}{\partial y} \right)^T.$$

The term $\nabla^2 \mathbf{u}$ is called the Laplacian; we reserve the symbol Δ for it and write (again in two dimensions)

$$\Delta \mathbf{u} \equiv \nabla^2 \mathbf{u} = \langle \nabla, \nabla \mathbf{u} \rangle = \left(\frac{\partial^2 \mathbf{u}}{\partial x^2} + \frac{\partial^2 \mathbf{u}}{\partial y^2} \right).$$

Last of these notation remarks: we will often abbreviate the partial derivative into the symbol ∂_t:

$$\partial_t \mathbf{u} \equiv \frac{\partial \mathbf{u}}{\partial t}.$$

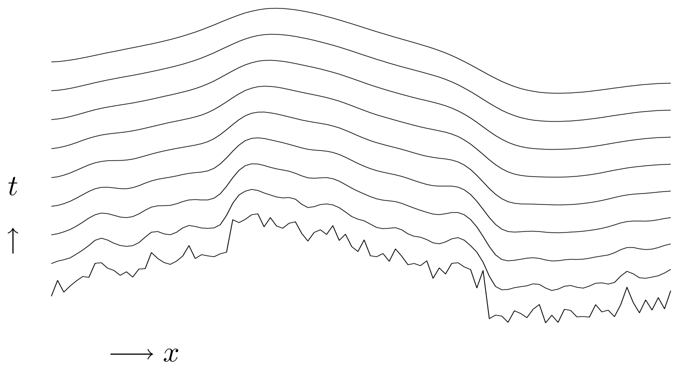


Figure 2.1: Linear scale-space representation: signal becomes smoother as the scale parameter t increases.

possesses the solution

$$\mathbf{u}(x,t) = \begin{cases} \mathbf{f}(x) & t = 0 \\ (G_{\sqrt{2t}} * \mathbf{f})(x) & t > 0. \end{cases} \tag{2.11}$$

We see that $\mathbf{u}(x,t)$ and $L(x,t)$ are identical, and the linear scale space can be formed either by convolving the input signal $\mathbf{f}$ with the Gaussian $G_{\sqrt{2t}}$, or by solving the partial differential equation (2.9).

Either way of the linear scale space creations fulfills all of the above mentioned requirements imposed on a scale space transformation T_t: for example, it never introduces new local extrema into the one-dimensional data during the smoothing process, it possesses the semi-group property. As the scale parameter t translates into the width of the smoothing kernel $G_{\sqrt{2t}}$, higher t means that more input points contribute significantly to the computation of an output point, and the output data become smoother. This important observation is illustrated in Figure 2.1.

However, the linear scale space has its limitations: whether we smooth uniformly by a rotationally symmetric Gaussian kernel, or diffuse the data equally in all directions, the process not only removes undesirable local extrema (noise) but also harms important features of the image, blurs and dislocates edges. To overcome these drawbacks, we have to move to nonlinear filters; nonlinear diffusion offers an excellent alternative.

2.3 Nonlinear diffusion

2.3.1 General formulation

Nonlinear diffusion has deservedly attracted much attention in the field of image processing for its ability to reduce noise while preserving (or even enhancing) important features of the image, such as edges or discontinuities; this can be opposed to linear diffusion (alias Gaussian filtering or linear scale-space representation) which not only removes noise but also blurs and dislocates edges. A good introduction to NL diffusion can be found e.g. in [40, 44] or [45].

Diffusion as a physical process equilibrates concentration differences without creating of destroying mass. This observation can be translated into the following:

- Fick's law
$$\psi = -\boldsymbol{D} \cdot \nabla \mathbf{u} \tag{2.12}$$
says that a concentration gradient $\nabla \mathbf{u}$ causes a flux ψ which aims to compensate this gradient; the relation between $\nabla \mathbf{u}$ and ψ is described by the diffusion tensor $\boldsymbol{D}$.[3]

- The continuity equation expresses the idea that mass can be only transported, not created or destroyed,
$$\partial_t \mathbf{u} = -\operatorname{div} \psi \tag{2.13}$$
where ∂_t represents differentiation with respect to time t and the divergence operator is equivalent to a scalar product of the gradient operator with the vector ψ, i.e. $\operatorname{div} \psi = \langle \nabla, \psi \rangle$.[4]

The two equations can be combined to yield the *diffusion equation*

$$\partial_t \mathbf{u} = \operatorname{div}(\boldsymbol{D} \cdot \nabla \mathbf{u}) \tag{2.14}$$

The simply looking equation (2.14) and other forms derived from it describe a large class of image processing operations. If we replace the diffusion tensor $\boldsymbol{D}$ by a single scalar value for the whole image domain, the process is equivalent to uniform smoothing and the linear scale space from the preceding section. If $\boldsymbol{D}$ itself depends on the evolving image $\mathbf{u}$, the resulting equation describes a *nonlinear diffusion* filter. We adopt the Weickert's terminology [45] and call the diffusion *anisotropic* if the flux ψ and $\nabla \mathbf{u}$ are not parallel. If $\psi \parallel \nabla \mathbf{u}$, the diffusion process is *isotropic* and the diffusion tensor can be replaced by a scalar-valued *diffusivity* g.

2.3.2 History

Nonlinear diffusion was first suggested in the pioneering work of Perona and Malik [28]. Motivated mainly by edge detection, they criticized the linear scale space for which the true location of a boundary at a fine scale is not directly available in

[3] The diffusion tensor $\boldsymbol{D}$ is generally a $N \times N$ square matrix different for each point of the data $\mathbf{u} \in \mathbb{R}^N$. Its role is basically to rotate and scale the gradient vector $\nabla \mathbf{u}$. More information on the diffusion tensor and its creation will be given in the succeeding paragraphs and in Section 2.4.

[4] The equation (2.13) formulates the conservation law in a local, differential form, omitting boundary conditions. The equivalent (and perhaps more intuitive) integral form can be obtained using the divergence theorem
$$\int_V \operatorname{div} \psi \, \mathrm{d}V = \int_{\partial V} \langle \psi, n \rangle \, \mathrm{d}r$$
where V is some region, ∂V its closed boundary, and n is a unit vector normal to this boundary.

the coarse scale image, edges are blurred and dislocated, and their tracking to finer scales is complicated and computationally expensive. They suggested to move to nonlinear scale space, formed under the following criteria for generating multiscale 'semantically meaningful' description of images:

- Causality: no spurious detail should be generated passing from finer to coarser scales.
- Immediate localization: boundaries should be sharp and coincide with the semantically meaningful boundaries at that resolution.
- Piecewise smoothing: intra-region smoothing should be preferred to inter-region smoothing.

To satisfy these criteria, they proposed to adapt the diffusion to the local image characteristics by introducing a space- and time-variant diffusion coefficient $c(x, y, t)$, and formulated the following nonlinear diffusion equation[5]

$$\partial_t \mathbf{u} = \operatorname{div}(c(x, y, t) \cdot \nabla \mathbf{u}) \tag{2.15}$$

with $\mathbf{u}$ the image, div the divergence operator and ∇ the gradient operator.

To make the filter prefer intra-region to inter-region smoothing and thus preserve edges, we need to know the region boundaries at each scale. Obviously, this information is not available a priori; the best we can do is to estimate the boundary location by an edge detector. Perona and Malik claim that the simplest $\nabla \mathbf{u}(x, y, t)$ works excellently. The conduction coefficient $c(x, y, t)$ is chosen as a function

$$c(x, y, t) = g(|\nabla \mathbf{u}(x, y, t)|) \tag{2.16}$$

with $g(\cdot)$ a nonnegative monotonically decreasing function with $g(0) = 1$ and $g(\infty) = 0$, for example

$$g(s) = \frac{1}{1 + (s/\lambda)^{1+\alpha}}, \ \alpha > 0. \tag{2.17}$$

Let $|\psi(s)| = s \cdot g(s)$ denote the flux magnitude[6]; it is nonmonotone, increasing for $s < \lambda$ and decreasing for $s > \lambda$ – see Figure 2.2 on page 19. A good choice of the parameters in g leads to the desirable result of blurring small discontinuities simultaneously with sharpening important edges. The parameter λ serves as a threshold of gradient size: a smaller gradient is diffused, positions of a larger gradient are treated as edges. However, the distinction is not crisp; the parameter α influences the steepness of the classification function g around λ.

Perona and Malik [28] prove the maximum–minimum principle for the discrete approximation of their diffusion equation. They suggest that the nonlinear diffusion

[5] Perona and Malik called their diffusion equation anisotropic. However, they used a scalar-valued diffusivity, the flux is parallel to the image gradient and the diffusion is isotropic in the terminology of Weickert [44, 45].

[6] This formula is consistent with the isotropic case of equation (2.12) if we choose $s = |\nabla \mathbf{u}|$.

filter can be applied to a variety of fields: it outperforms the widely used Canny approach to edge detection, produces clear edges even without non-maximum suppression, preserves junctions, is simply formulated and the edges can be computed in parallel. Another application can be the image restoration and segmentation. Perona and Malik compare their filter to the approaches using the minimization of some energy functional which evaluates the quality of the filtering result (Mumford and Shah [24], Blake and Zisserman [8]) and show that the nonlinear diffusion may be seen as a gradient descent for a suitable functional minimization. We return to this point in Section 2.3.3.

The publications which followed on the Perona-Malik's paper were numerous; we recommend the reader to consult the Weickert's book [45] for a very good overview. We mention only several of the most important ideas in the following paragraphs.

First to a theoretical problem: Catté, Lions, Morel and Coll [10] showed that the continuous Perona–Malik model is ill posed; very close pictures can produce divergent solutions and therefore – in the context of edge detection – very different edges. This is caused by the fact that the diffusivity g used in [28] leads to flux $s \cdot g(s)$ decreasing for some s and the scheme may work locally like the inverse heat equation which is known to be ill posed, and can develop singularities of any order in arbitrarily small time. This possible misbehaviour surely represents a severe drawback of the Perona–Malik model when applied to data effected by noise. However, as Catté *et al.* mention and Weickert and Benhamouda [50] study in more detail, discrete implementations work as a regularization factor by introducing implicit diffusion into the model, and the filter is usually observed to be stable (with staircasing effect as the only observable instability: a single ramp-like edge may lead to several steps of piecewise constant values).

To solve the problem correctly in the continuous settings, Catté *et al.* propose a new model with the only modification of replacing the gradient $|\nabla \mathbf{u}|$ in equation (2.15) by its estimate $|\nabla G_\sigma * \mathbf{u}|$ where G_σ can be any smoothing kernel, the Gaussian represents a classical example[7]. With this simple modification, Catté *et al.* were able to obtain the proofs of existence and uniqueness of the solution for any $\sigma > 0$, so the new equation

$$\partial_t \mathbf{u} = \operatorname{div}(g(|\nabla \mathbf{u}_\sigma|) \cdot \nabla \mathbf{u}) \tag{2.18}$$

represents a correct basis for a nonlinear scale space. We will employ results of Catté *et al.* when giving a complete isotropic NL diffusion formulation together with its discretization in Section 2.6.

The second problem with the Perona–Malik (as well as with the Catté *et al.*) isotropic model is a practical one: as the diffusion is stopped near strong edges, not only the desired information but also the noise will be left untouched there.

[7] Note that the differentiation and the convolution commute,

$$(\nabla G_\sigma) * \mathbf{u} = \nabla (G_\sigma * \mathbf{u}).$$

We will denote either of these expressions by $\nabla \mathbf{u}_\sigma$ in the following text.

To provide a solution, we have to turn our attention to anisotropic filters where the flux is not generally parallel to the image gradient. Anisotropic diffusion filters have been studied thoroughly by Weickert [43, 45, 48, 46, 47]. The same author can be also praised for the analysis of semidiscrete (i.e. with sampled data but continuous time parameter) and fully discrete versions of the nonlinear diffusion scale spaces [42, 45]. We include some information on the nonlinear scale-space properties in Section 2.3.4; anisotropic nonlinear diffusion forms the main topic of Section 2.4.

2.3.3 Link to other methods

We mentioned in the introduction that image restoration using nonlinear diffusion was created at an intersection of many strong fields such as scale space, energy minimization, modeling of physical phenomena, robust statistics, and mathematical morphology. In addition to the points discussed above, we mention in this section several of the most pertinent connections to other methods, needed for a better understanding of the filtering procedures. The list will be by no means exhaustive. The interested reader is referred to the Weickert's book [45] for an excellent and abundant survey of diffusion and adjacent fields; we neither intend nor dare to equal the 453 references cited there.

Variational formulation

Many authors dealing with the ill posed tasks of edge detection or reconstruction of corrupted data choose to regularize the problem by turning it into minimization of a given energy functional. In this section we consider an approach which illustrates very well the connection between such a variational formulation and the algorithms for nonlinear diffusion.

Nordström [25][8] suggests to solve the global regularization and edge detection problem by minimizing the energy functional

$$E_{\mathbf{f}}(\mathbf{u},\mathbf{w}) = \int_\Omega \big(\beta\cdot(\mathbf{u}-\mathbf{f})^2 + \mathbf{w}\cdot|\nabla\mathbf{u}|^2 + \lambda^2\cdot(\mathbf{w}-\ln\mathbf{w})\big)\,\mathrm{d}x \tag{2.19}$$

where the parameters β, λ are positive weights and $\mathbf{w}\colon \Omega \to (0,1]$ is a smooth continuity control function which can also be understood as a fuzzy edge representation: $\mathbf{w}$ approaches 1 in the interior of a region; at edges, $\mathbf{w}$ is close to 0. The first of the summands of $E_{\mathbf{f}}$ punishes deviations of the solution $\mathbf{u}$ from the input data $\mathbf{f}$ *(deviation cost)*, the other two measure the ‘quality’ of the solution: the middle one is the *stabilizing cost* favouring smoothness of $\mathbf{u}$ within continuous regions, and the last, *edge cost*, is intended to penalize edges in order to prevent pathological solutions with edges filling a large part of the image domain.

Minimum of the functional $E_{\mathbf{f}}$ is attained where its first variation is zero; this

[8] We changed the notation to that of Weickert [45] to keep it consistent with other sections.

basic rule of the calculus of variations yields the following Euler equations:

$$0 = \beta \cdot (\mathbf{u} - \mathbf{f}) - \operatorname{div}(\mathbf{w}\, \nabla \mathbf{u}) \qquad \text{on } \Omega \tag{2.20}$$

$$0 = \lambda^2 \left(1 - \frac{1}{\mathbf{w}}\right) + |\nabla \mathbf{u}|^2 \qquad \text{on } \Omega \tag{2.21}$$

$$0 = \mathbf{w}\, \frac{\partial \mathbf{u}}{\partial n} \qquad \text{on } \partial\Omega \tag{2.22}$$

where n is the direction perpendicular to the image boundary $\partial\Omega$. The last equation (2.22) expresses the Neumann boundary conditions: the derivative of $\mathbf{u}$ in the direction of n is zero, i.e. the image is reflected at the boundary and the image region exchanges no flow with the exterior.

Solving (2.21) for $\mathbf{w}$ gives

$$\mathbf{w} = \frac{1}{1 + \frac{|\nabla \mathbf{u}|^2}{\lambda^2}} \tag{2.23}$$

and we can see that $\mathbf{w}$ is identical with a special case of the Perona-Malik diffusivity $g(|\nabla \mathbf{u}|)$ from equation (2.17). Therefore, the equation (2.20) can be regarded as the steady state of the *biased diffusion equation*

$$\partial_t \mathbf{u} = \operatorname{div}(g(|\nabla \mathbf{u}|) \cdot \nabla \mathbf{u}) + \beta \cdot (\mathbf{f} - \mathbf{u}). \tag{2.24}$$

The diffusion-reaction equation (2.24) consists of the Perona-Malik process with an additional bias term $\beta \cdot (\mathbf{f} - \mathbf{u})$. Nordström stresses that his biased diffusion leads to nontrivial steady-state solutions and thus frees the user from setting a finite stopping time T. However, we support the Weickert's argument [45] that the Nordström's formulation just shifts the problem of setting T to the problem of determining the weighting factor β. The biased diffusion with a nonzero β is essentially equivalent to the unbiased diffusion of Perona and Malik with a finite T, so it is only a matter of personal taste which of the two approaches to prefer. Moreover, the functional (2.19) is nonconvex so the biased diffusion is likely to converge towards its local minimum.

Link to robust statistics

Black and Sapiro [7, 6] offer an interesting insight into nonlinear diffusion from the viewpoint of robust statistical estimation. They demonstrate that the Perona-Malik equation is equivalent to a robust procedure that estimates a piecewise constant function from noisy input data.

Let us form a statistical model of the image: piecewise constant function is corrupted by zero-mean Gaussian noise with small variance. Consider the image intensity differences, $I_p - I_s$, between the pixel s and its neighbouring pixels p. Within one constant image region, the neighbour differences will be small, zero-mean and normally distributed. An optimal estimator of the 'true' value at pixel s minimizes the sum of squares of the neighbour differences, which is attained if I_s is set to be the mean of the neighbouring pixels.

If there is an edge, boundary or discontinuity near pixel s, the differences will not be normally distributed as the neighbours are sampled from two different distributions. To prevent the estimate at such a position from being distorted, the robust statistical procedures assign smaller weights to the (large) neighbour differences $I_p - I_s$ which can be viewed as *outliers*. The choice of the function which would downweight or reject outliers is critical for the method.

Black and Sapiro state that the Perona–Malik stopping function

$$g(s,\lambda) = \frac{2}{2 + \frac{s^2}{\lambda^2}} \tag{2.25}$$

is after multiplication by s (to obtain the flux function $|\psi|$) proportional to the influence function used in robust statistics. The flux after integration is proportional to the *Lorentzian* error norm. As a result, outliers have little influence on the solution.

The authors compare the Perona–Malik g to a stopping function derived from the Tukey's biweight, which they claim is more robust and leads to sharper boundaries[9]:

$$g(s,\lambda) = \begin{cases} \frac{1}{2}\left(1 - (\frac{s}{\lambda \cdot \sqrt{5}})^2\right)^2 & |s| \leq \lambda \cdot \sqrt{5} \\ 0 & \text{otherwise} \end{cases} \tag{2.26}$$

Using this diffusivity, the outliers (i.e. the data separated from the current position by an edge) will have *no* influence on the solution. The Perona-Malik and the Tukey diffusivity functions are plotted in Figures 2.2 and 2.4, respectively.

Adopting the robust statistical terminology, edges can be defined as outliers of some local image statistics, and detected simply by considering the locations where the diffusivity function g approaches zero.

2.3.4 Theoretical properties

In [45], Weickert proved many useful properties of the nonlinear diffusion under some general assumptions about the input function $\mathbf{f}$ and some conditions imposed on the diffusion tensor $\boldsymbol{D}$. More precisely, Weickert studies the continuous anisotropic model

$$\partial_t \mathbf{u} = \operatorname{div}(\boldsymbol{D}(\nabla \mathbf{u}_\sigma) \cdot \nabla \mathbf{u}) \qquad \text{on } \Omega \times (0,T], \tag{2.27}$$

$$\mathbf{u}(x,0) = \mathbf{f}(x) \qquad \text{on } \Omega, \tag{2.28}$$

$$\langle (\boldsymbol{D}(\nabla \mathbf{u}_\sigma) \cdot \nabla \mathbf{u}), n \rangle = 0 \qquad \text{on } \partial\Omega \times (0,T] \tag{2.29}$$

where $\sigma, T > 0$, and the diffusion tensor[10] $\boldsymbol{D} = \left(\begin{smallmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{smallmatrix}\right)$, itself a function of the image gradient, $\boldsymbol{D}(\nabla \mathbf{u}_\sigma)$, satisfies the following properties:

[9] We added the constant $\sqrt{5}$ to the fraction denominator of the original Black–Sapiro's equation to keep the formula consistent with other diffusivities: we require that the flux function $s \cdot g(s,\lambda)$ starts to decrease at λ.

[10] Weickert was mainly interested in anisotropic NL diffusion with a diffusion tensor $\boldsymbol{D}$; however, the results transfer directly to the isotropic case which uses the scalar diffusivity g. Clearly, the

(C1) Smoothness : $\boldsymbol{D}$ depends continuously and smoothly on its parameter $\nabla \mathbf{u}_\sigma$.

(C2) Symmetry : $d_{12} = d_{21}$.

(C3) Uniform positive definiteness : there exists a positive lower bound for the eigenvalues of $\boldsymbol{D}$, and the bound is uniform (i.e. independent of the argument).

We mention several of the results obtained under these assumptions.

- Well-posedness (anisotropic extension of the proof by Catté et al. [10]): the equations (2.27)–(2.29) have a unique solution $\mathbf{u}(x,t)$ which depends continuously on the input image $\mathbf{f}$.
- Extremum principle:
$$\inf_{x\in\Omega} \mathbf{f}(x) \leq \mathbf{u}(x,t) \leq \sup_{x\in\Omega} \mathbf{f}(x). \tag{2.30}$$
- Average grey level invariance: the average grey level μ is preserved by the diffusion filtering,
$$\frac{1}{|\Omega|}\int_\Omega \mathbf{u}(x,t)\,\mathrm{d}x = \mu, \qquad \forall t. \tag{2.31}$$
- Nonenhancement of local extrema: if $\mathbf{u}(\cdot,t)$ has a local extremum with a nonzero Hessian at x, then
$$\partial_t \mathbf{u}(x,t) < 0 \qquad \text{if } x \text{ is a local maximum,} \tag{2.32}$$
$$\partial_t \mathbf{u}(x,t) > 0 \qquad \text{if } x \text{ is a local minimum.} \tag{2.33}$$
- Information-reduction, smoothing:
  - all even central moments,
  $$\frac{1}{|\Omega|}\int_\Omega (\mathbf{u}(x,t)-\mu)^{2n}\,\mathrm{d}x, \qquad n\in\mathbb{N}, \tag{2.34}$$
  are decreasing for $t\in[0,\infty)$;
  - the energy $\|\mathbf{u}(t)\|^2_{L^2(\Omega)}$ is decreasing with t;
  - the Shanon-Wiener entropy
  $$S_1(\mathbf{u}(x,t)) = -\int_\Omega \mathbf{u}(x,t)\ln(\mathbf{u}(x,t))\,\mathrm{d}x \tag{2.35}$$
  is increasing with t (see also [37, 38] for details and a study of the behaviour of generalized entropies in a scale space).

following equality holds:
$$g\cdot\nabla u = g\cdot \boldsymbol{I}\cdot\nabla u = \begin{pmatrix} g & 0\\ 0 & g\end{pmatrix}\cdot\nabla u$$
and the isotropic 'diffusion tensor' has the form $\boldsymbol{D} = \left(\begin{smallmatrix} g & 0\\ 0 & g\end{smallmatrix}\right)$.

∘ the diffusion converges to a constant,

$$\lim_{t\to\infty} \mathbf{u}(x,t) = \mu, \tag{2.36}$$

and the convergence is uniform on the closure of the function domain, $\overline{\Omega}$.

There exist discretization schemes of the (isotropic or anisotropic) NL diffusion for which the properties mentioned above carry over to the semidiscrete (continuous time t, discrete space) or fully discrete situations. The latter case is the most common in practice: an image is discretized and represented by values $f_{x,y}$, $x = 1,\ldots,N_1$, $y = 1,\ldots,N_2$, and the diffusion is computed for a discrete set of scales t_k. We will denote by $\mathbf{u}^k$ the discrete solution of the diffusion equation at time t_k. For sake of notation simplicity, a discrete image will be sometimes regarded as a vector $\mathbf{f} \in \mathbb{R}^N$ where $N = N_1 \cdot N_2$ is the total number of pixels, and the image elements are f_j, $j \in J \equiv \{1,\ldots,N\}$.

Still in [45], Weickert studies the following discrete nonlinear diffusion model:

$$\mathbf{u}^0 = \mathbf{f} \tag{2.37}$$

$$\mathbf{u}^{k+1} = \boldsymbol{Q}(\mathbf{u}^k) \cdot \mathbf{u}^k, \qquad \forall k \in \mathbb{N}_0 \tag{2.38}$$

where the $N \times N$ diffusivity matrix $\boldsymbol{Q} = (q_{ij})$ has the following properties:

(D1) $\boldsymbol{Q}$ depends continuously on the data $\mathbf{u}$: $\boldsymbol{Q} \in C(\mathbb{R}^N, \mathbb{R}^{N\times N})$,

(D2) symmetry: $q_{ij} = q_{ji}$,

(D3) unit row sum: $\sum_{j\in J} q_{ij} = 1$,

(D4) nonnegativity: $q_{ij} \geq 0$,

(D5) irreducibility: any two pixels can be connected by a path of nonvanishing diffusivities. Formally:

$$\forall i,j \in J \;\; \exists l_0,\ldots,l_r \in J,\; l_0 = i,\; l_r = j : \qquad q_{l_p l_{p+1}} \neq 0, \qquad \forall p = 0,\ldots,r-1. \tag{2.39}$$

(D6) positive diagonal: $q_{ii} > 0$.

For this model, Weickert proves discrete counterparts of the continuous diffusion properties; we list a similar selection as above.

- extremum principle:

$$\min_{j\in J} f_j \leq u_i^k \leq \max_{j\in J} f_j \qquad \forall i \in J,\;\; \forall k \in \mathbb{N}_0 \tag{2.40}$$

where u_i^k is the i-th element of the solution at time k.

- the average grey value $\mu = \frac{1}{N}\sum_{j\in J} f_j$ is preserved by the discrete diffusion filter,

$$\frac{1}{N}\sum_{j\in J} u_j^k = \mu, \qquad \forall k \in \mathbb{N}_0. \tag{2.41}$$

- Information-reduction, smoothing:
  - all even central moments,

    $$\frac{1}{N}\sum_{j\in J}(u_j^k - \mu)^{2n}, \qquad n \in \mathbb{N}, \tag{2.42}$$

    are decreasing in k;
  - the energy $\|\mathbf{u}^k\|_p = \left(\sum_{j\in J}\left(u_j^k\right)^p\right)^{\frac{1}{p}}$ is decreasing in k for all $p \geq 1$;
  - the entropy

    $$S(\mathbf{u}^k) = -\sum_{j\in J} u_j^k \ln(u_j^k) \tag{2.43}$$

    is increasing with k if $\min_{j\in J} f_j > 0$.
  - the diffusion converges to a constant,

    $$\lim_{k\to\infty} u_j^k = \mu, \qquad \forall j \in J. \tag{2.44}$$

Numerical algorithms which approximate the continuous diffusion process and fall into the discrete scheme (2.37)–(2.38), thus possessing all the properties mentioned in the previous paragraph, are explored in Section 2.6.

2.3.5 The diffusion parameters

The meaning and choice of the parameters for nonlinear diffusion is discussed in this section.

Noise scale σ

The parameter σ denotes the width of the Gaussian kernel used to pre-smooth the image gradient before the diffusivity function computation. It is also called *noise scale*: by smoothing by G_σ, the noise of scale smaller than σ will not influence the diffusion results. Note also that the discretization artefacts are mitigated by the pre-smoothing.

If the noise in neighbouring pixels is uncorrelated, the value $\sigma = \Delta x$ (where Δx is the grid size; we usually assume $\Delta x = 1$) is sufficient for a large interval of noise variances, and is used exclusively in our experiments.

Diffusivity function g

The diffusivity function $g\colon s \to [0,1]$ is a nonincreasing function of either the size of the image gradient, $s = |\nabla u|$, or its regularized version, the smoothed gradient $s = |\nabla u_\sigma|$. The function $g(s)$ plays a role of a 'fuzzy detector' of the presence of an edge at a particular position: if s is small, there is a minor probability of an edge at that position, and g is close to 1; if, on the other hand, s is large, the location is likely to belong to an edge, and the value of g will be close to zero.

The adjectives 'small' or 'large' for s in the previous paragraph have to be understood relative to the edge threshold parameter λ. All the diffusivity functions we will mention are constructed so that the flux magnitude $|\psi(s)| = s \cdot g(s)$ is increasing on the interval $[0, \lambda]$ and decreasing (or at least nonincreasing) on $[\lambda, \infty)$. This way, the areas with $s \ll \lambda$ are considered as flat regions where the diffusivity is allowed; at the locations of $s \gg \lambda$, the (isotropic) NL diffusion is virtually stopped and the diffusion process behaves locally like a backwards heat equation with the effect of edge enhancement.

Quite a few formulations of the diffusivity function have appeared in the literature. The original Perona-Malik's formulation was

$$g(s) = \frac{C}{1 + (s/\lambda)^{1+\alpha}}, \ \alpha > 0, \tag{2.45}$$

an example with $\alpha = 1$ (and $C = 1$) is drawn together with its flux function in Figure 2.2.

Black and Sapiro [7] link the Perona-Malik diffusivity to the weighting functions of robust statistical estimation — see page 13. The alternative diffusivity function of equation (2.26), developed from the Tukey's biweight, is shown in Figure 2.4. Note that being identically equal to zero for $s > \lambda \cdot \sqrt{5}$, their diffusivity function does not satisfy the criteria for nonlinear scale spaces stated in Section 2.3.4.

Weickert [45, 53] uses the diffusivity function of the following type for his isotropic and edge-enhancing methods:

$$g(s) = 1 - \exp\left(\frac{c}{(s/\lambda)^m}\right). \tag{2.46}$$

In our experiments, we use this formula with the case $m = 4$ and the constant $c \doteq -2.33667$ determined so that the flux $s \cdot g(s)$ is increasing for $s < \lambda$ and decreasing for $s > \lambda$.[11] This diffusivity formulation combines the theoretical advantages of a diffusivity positive everywhere with a fast decline of g around λ, implying that strong edges are well preserved for a long time period of the diffusion process.

Diffusivity parameter λ

We explained the role of the parameter λ in the preceding paragraph. This section discusses how to determine its value.

[11] Weickert used a different constant c in his original formulation. The difference arises from the fact that Weickert computes the diffusivity from the *squared* gradient size, $g(|\nabla \mathbf{u}_\sigma|^2)$; we keep the equation consistent with other sections and use $s = |\nabla \mathbf{u}_\sigma|$.

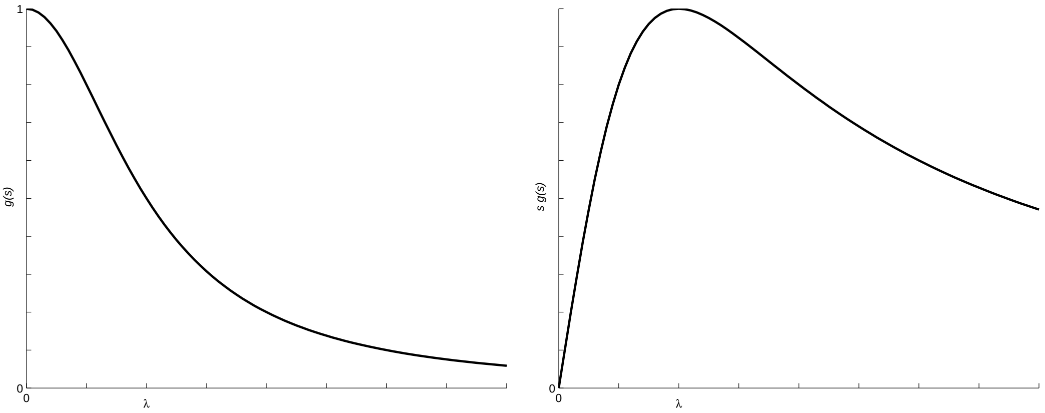


Figure 2.2: The Perona–Malik diffusivity $g(s) = \frac{1}{1+(s/\lambda)}$ (left); the corresponding flux function $s \cdot g(s)$ (right).

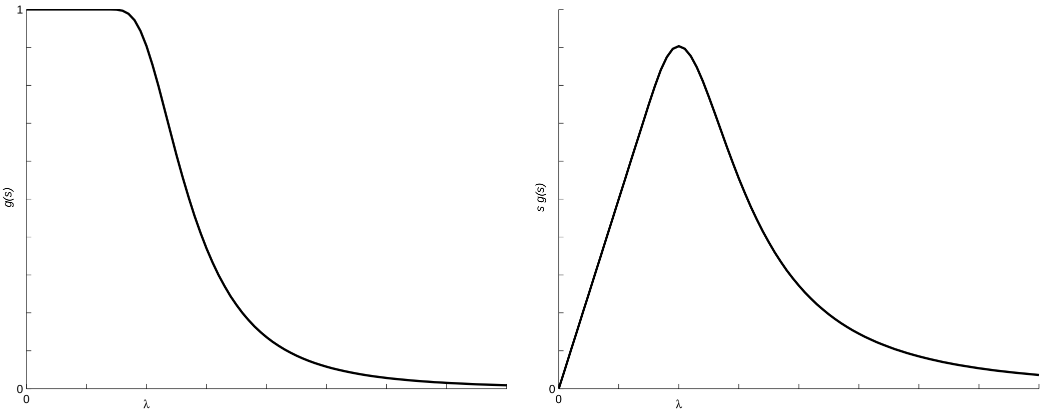


Figure 2.3: The Weickert diffusivity $g(s) = 1 - \exp\left(\frac{-2.33667}{(s/\lambda)^4}\right)$ (left); the corresponding flux function $s \cdot g(s)$ (right).

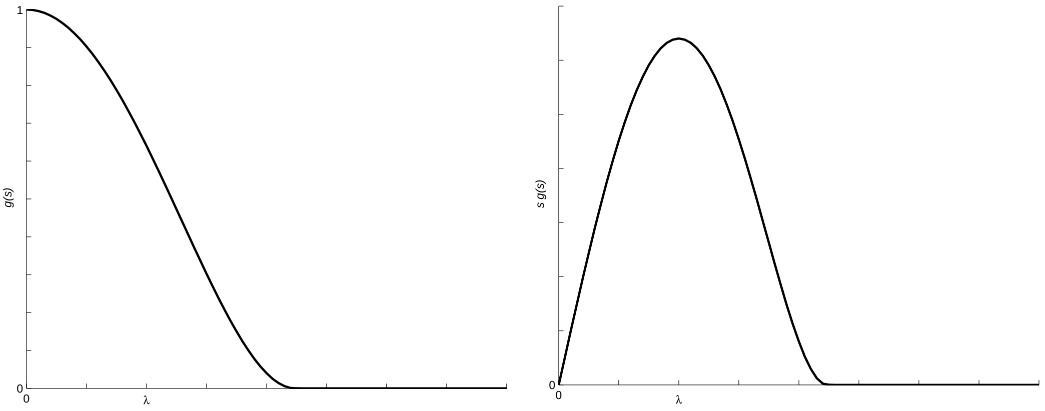


Figure 2.4: The Black–Sapiro diffusivity $g(s) = \frac{1}{2}\left(1 - \left(\frac{s}{\lambda\cdot\sqrt{5}}\right)^2\right)^2$ for $s \leq \lambda \cdot \sqrt{5}$, $g(s) = 0$ otherwise (left); the corresponding flux function $s \cdot g(s)$ (right).

Perona–Malik [28] suggest two possibilities how to choose λ: set a fixed value by hand, or exploit the idea of Canny's noise estimator and set λ equal to some percentile p of the image gradients at each diffusion iteration (e.g. 90%; we will write it as a decimal number $p \in [0,1]$). This way a chosen $(1-p)$ percentage of the strongest edges (given by an area of the image) remains almost unharmed by the diffusion filter, and the approach has two advantages: firstly, choosing some percentile of the image to be treated as edges is more natural and intuitive than having to decide directly on a fixed value for λ. Secondly, recomputing the percentile in each diffusion step leads to a fast decrease in λ and has a stabilizing effect on the diffusion results (although theoretically the procedure still converges to a globally constant function).

In [7, 6], Black and Sapiro suggest to globally or locally estimate λ by a procedure from robust statistics [32]. What they call 'robust scale' is computed as

$$\lambda = 1.4826 \text{ median}\big(\, \big| |\nabla \mathbf{f}| - \text{median}(|\nabla \mathbf{f}|) \big| \,\big) \tag{2.47}$$

where $\mathbf{f}$ is the image and the constant in the formula comes from the fact that the median absolute deviation[12] of a normal distribution with unit variance is approximately $0.6745 = 1/1.4826$.

The paper [6] adds a spatially varying λ. The authors compute the robust scale estimate λ_{el} using equation (2.47) in a $n \times n$ window, and define the spatially varying λ_l as the maximum of this local and the global one:

$$\lambda_l = \max(\lambda, \lambda_{el}) \tag{2.48}$$

The locally adapted λ_l may be useful for simplification and filtering of spatially varying textures: in areas of high contrast, larger gradient is needed to consider a pixel as belonging to an edge; the global estimate as a lower bound prevents the amplification of noise in the low contrast areas. Then it is possible to detect edges robustly by detecting the points which are treated as outliers by the given function g.

Stopping time T

The stopping time T has a strong effect on the diffusion result. Its choice has to balance two contradictory motivations: small T gives more trust to the input data (and leaves more details and noise in the data unfiltered), while large T means that the result becomes dominated by the (piecewise) constant model which is inherent in the diffusion equations. The scale-space people often set T to a large value (ideally infinity) and observe how the diffused function evolves with time (and converges to a constant value). As we are more concerned with image restoration and we want to obtain nontrivial results from the diffusion filter, we will have to pick a single (finite) time instant T and stop the diffusion evolution there. We cite here the approaches to stopping time selection which have appeared in the literature, and comment on them.

[12] The median absolute deviation of f is defined as $\text{median}(|f - \text{median}(f)|)$.

Catté *et al.* [10] argue that the regularization by G_σ for the diffusivity computation introduces a sort of time: the result of convolution is the same as the solution to the linear heat equation at time $t = \frac{\sigma^2}{2}$, so it is coherent to correlate the stopping time T and the 'time' of the linear diffusion. However, the equality $t = \frac{\sigma^2}{2}$ is rather a lower estimate of the stopping time: because of the diffusion process inhibited near the edges, the nonlinear diffusion is always slower than the linear one, and needs a longer time to reach the desired results.

Dolcetta and Ferretti [12] recently formulated the time selection problem as a minimization of the functional

$$E(T) = \int_0^T E_c + E_s \tag{2.49}$$

where E_c is the computing cost and E_s the stopping cost, the latter encouraging filtering for small T. The authors provide a basic example

$$E_c = c \tag{2.50}$$

$$E_s = -\left(\int_\Omega |\mathbf{u}(x,T) - \mathbf{u}(x,0)|^2 \,\mathrm{d}x\right)^2 \tag{2.51}$$

where the constant c balancing the influence of the two types of costs has to be computed from a typical image to be filtered.

Sporring and Weickert in [38] study the behaviour of generalized entropies, and suggest that the intervals of minimal entropy change indicate especially stable scales with respect to evolution time. They estimate that such scales could be good candidates for stopping times in nonlinear diffusion scale spaces. However, as the entropy can be stable on whole *intervals*, it may be difficult to decide on a single stopping instant from that interval; we are unaware of their idea being brought to practice in the field of image restoration.

Weickert mentioned more ideas on the stopping time selection, more closely linked to the noise-filtering problem, in [47]. They are based on the notion of relative variance.

Let $\mathrm{var}(\mathbf{u}(t))$ denote the variance of an image $\mathbf{u}(t)$. This variance is monotonically decreasing with t and converges to zero as $t \to \infty$. The *relative variance*

$$r(\mathbf{u}(t)) = \frac{\mathrm{var}(\mathbf{u}(t))}{\mathrm{var}(\mathbf{u}(0))} \tag{2.52}$$

decreases monotonically from 1 to 0 and can be used to measure the distance of $\mathbf{u}(t)$ from the initial state $\mathbf{u}(0)$ and the final state $\mathbf{u}(\infty)$. Prescribing a certain value for $r(\mathbf{u}(T))$ can therefore serve as a criterion for selection of the stopping time T.

Let again $\widetilde{\mathbf{f}}$ be the ideal data, the measured noisy image $\mathbf{f} = \widetilde{\mathbf{f}} + n$, and let the noise n be of zero mean and uncorrelated with $\widetilde{\mathbf{f}}$. Now assume that we know the variance of the noise, or (equivalently, on the condition that the noise and the signal are uncorrelated) the *signal-to-noise ratio*, defined as the ratio between the original image variance and the noise variance,

$$\mathrm{SNR} \equiv \frac{\operatorname{var}\left(\widetilde{\mathbf{f}}\right)}{\operatorname{var}(n)}. \tag{2.53}$$

As the signal $\widetilde{\mathbf{f}}$ and the noise n are uncorrelated, we have

$$\operatorname{var}(\mathbf{f}) = \operatorname{var}\left(\widetilde{\mathbf{f}}\right) + \operatorname{var}(n). \tag{2.54}$$

Substituting from this equality for $\operatorname{var}(n)$ into (2.53), we obtain by simple rearrangement that

$$\frac{\operatorname{var}\left(\widetilde{\mathbf{f}}\right)}{\operatorname{var}(\mathbf{f})} = \frac{1}{1 + \frac{1}{\mathrm{SNR}}}. \tag{2.55}$$

We take the noisy image for the initial condition of our diffusion filter, $\mathbf{u}(0) = \mathbf{f}$. An ideal diffusion filter would first eliminate the noise before significantly affecting the signal; if we stop at the right moment, we might substitute the filtered data $\mathbf{u}(T)$ for the ideal signal $\widetilde{\mathbf{f}}$ in (2.55). Relying on this analogy, we can choose the stopping time T such that the relative variance satisfies

$$r(\mathbf{u}(T)) = \frac{\operatorname{var}(\mathbf{u}(T))}{\operatorname{var}(\mathbf{u}(0))} = \frac{1}{1 + \frac{1}{\mathrm{SNR}}} \tag{2.56}$$

Weickert remarks that the criterion (2.56) tends to underestimate the optimal stopping time, as even a well-tuned filter cannot avoid influencing the signal before eliminating the noise.

So far the Weickert's suggestions from [47]: knowing the SNR, the formula (2.56) tells us when to stop the diffusion. However, our experiments indicate that this approach does not usually yield the optimal stopping time; we study the situation in more detail and develop a novel and reliable time-selection strategy based on signal–noise decorrelation in Section 3.1.

2.4 Anisotropic NL diffusion

Isotropic nonlinear diffusion with a scalar diffusivity g is stopped near the object boundary; it preserves the important edges/discontinuities in the data, but also leaves the noise near such positions unfiltered. To mitigate this undesirable effect, Weickert [43, 45, 46] proposes to make the amount of diffusion dependent not only on the position in the image, but vary it also between various directions at a single location. The process can be designed so that the smoothing perpendicular to the image gradient, i.e. along coherent structures (such as edges or lines) is preferred to smoothing across edges. To obtain this behaviour, the flux cannot be parallel to the image gradient (as is the case with the ‘classical’, isotropic NL diffusion equation), the diffusivity controlling the process is not a scalar any more, but a matrix, $\boldsymbol{D}$, leading to a more general equation of the *anisotropic nonlinear diffusion*

$$\partial_t \mathbf{u} = \operatorname{div}\left(\boldsymbol{D}(\nabla \mathbf{u}_\sigma) \cdot \nabla \mathbf{u}\right) \tag{2.57}$$

with the initial and boundary conditions, respectively,

$$\mathbf{u}(x,0)=\mathbf{f}(x), \qquad \langle \boldsymbol{D}(\nabla\mathbf{u}_\sigma)\cdot\nabla\mathbf{u}, n\rangle = 0 \ \text{ on } \partial\Omega \tag{2.58}$$

where n denotes the normal to the image boundary $\partial\Omega$. The diffusion starts from the input image $\mathbf{f}$ at $t=0$, and the boundary condition expresses the fact that no flux should pass through the image boundary.

The *diffusion tensor* $\boldsymbol{D}(\nabla\mathbf{u}_\sigma)$ and the way it is constructed have crucial influence on the properties of the resulting method, leading e.g. to edge-enhancing or coherence-enhancing procedures [45]. Whereas the latter can be used to close interrupted line-like structures, the former one, the edge-enhancing diffusion, and similar methods derived from it, are of importance for our task of noise reduction.

In this section we review the Weickert's suggestion to construct the diffusion tensor $\boldsymbol{D}$ from a structure tensor, and propose a minor alternation to make the method more suitable for the image restoration task.

2.4.1 Structure tensor

The strength of NL diffusion relies on its adaptation to the local image properties (such as gradient magnitude and orientation, or coherence of the structures). One possibility how to describe the local image information is to employ the structure tensor (see p. 56 of the Weickert's book [45] or again the paper [46], and references there).

Convolving an image $\mathbf{u}$ with a Gaussian kernel G_σ (where σ stands for the kernel variance, or *noise scale*), we obtain a smoothed version of the image, $\mathbf{u}_\sigma = G_\sigma * \mathbf{u}$. The gradient $\nabla\mathbf{u}_\sigma$ points in the direction of the highest intensity change in the smoothed image.

In order to identify gradients with the same orientation but of opposite directions, the gradient can be reconsidered in a matrix framework. Let us form a matrix $\boldsymbol{S}_0$ by a tensor product of the gradient vectors,

$$\boldsymbol{S}_0 = \nabla\mathbf{u}_\sigma \otimes \nabla\mathbf{u}_\sigma = \nabla\mathbf{u}_\sigma\, \nabla\mathbf{u}_\sigma^T. \tag{2.59}$$

The matrix $\boldsymbol{S}_0$ possesses an orthonormal basis of eigenvectors $\vec{v}_1, \vec{v}_2$ with $\vec{v}_1 \| \nabla\mathbf{u}_\sigma$ and $\vec{v}_2 \perp \nabla\mathbf{u}_\sigma$. The respective eigenvalues $|\nabla\mathbf{u}_\sigma|^2$ and 0 give the contrast (the squared gradient) in the eigendirections.

The information contained in the matrix $\boldsymbol{S}_0$ is already sufficient to control a combined diffusion (of different amounts) in the directions parallel and perpendicular to the gradient, so as to remove the small-scale noise both in homogeneous regions and at the edges without blurring the discontinuities in image intensity; we will return to this point later on. However, the local information cannot provide enough clues if we were after coherence enhancement, e.g. the restoration of interrupted line-like structures as in [46]. In such a case, information from some neighbourhood has to be assembled. This can be attained by a componentwise convolution of the matrix $\boldsymbol{S}_0$ with a Gaussian G_ϱ, where the *integration scale* ϱ

should reflect the characteristic window size over which the orientation is to be analysed:

$$\boldsymbol{S}_\varrho(\nabla \mathbf{u}_\sigma) = G_\varrho * (\nabla \mathbf{u}_\sigma \otimes \nabla \mathbf{u}_\sigma) \qquad (\varrho \geq 0) \tag{2.60}$$

Weickert recalls several names of the matrix $\boldsymbol{S}_\varrho$ from the literature: *structure tensor*, interest operator, scatter matrix, second moment matrix. The matrix is symmetric, $\boldsymbol{S}_\varrho = \left(\begin{smallmatrix} s_{11} & s_{12} \\ s_{12} & s_{22} \end{smallmatrix}\right)$, positive semidefinite, possesses orthonormal eigenvectors $\vec{v}_1, \vec{v}_2$ with

$$\vec{v}_1 \Big\| \begin{pmatrix} 2s_{12} \\ s_{22} - s_{11} + \sqrt{(s_{11} - s_{22})^2 + 4s_{12}^2} \end{pmatrix} \tag{2.61}$$

and the corresponding eigenvalues μ_1, μ_2,

$$\mu_{\{1,2\}} = \frac{1}{2}\left(s_{11} + s_{22} \pm \sqrt{(s_{11} - s_{22})^2 + 4s_{12}^2}\right) \tag{2.62}$$

where the plus sign belongs to μ_1. The eigenvalues describe the average, integrated contrast in the eigendirections; since $\mu_1 \geq \mu_2 \geq 0$, $\vec{v}_1$ is the orientation of the highest grey-value fluctuations, whereas $\vec{v}_2$ gives the prevailing local orientation, or coherence direction. Furthermore, the eigenvalues can be used to analyse the local structure: $\mu_1 = \mu_2 = 0$ in constant areas, straight edges lead to $\mu_1 \gg \mu_2 = 0$, corners can be identified by $\mu_1 \geq \mu_2 \gg 0$, and the expression $(\mu_1 - \mu_2)^2$ may serve as a measure of coherence of a structure. These properties can be exploited to control the amount of diffusion in the coherence direction.

2.4.2 From structure to diffusion tensor

The information contained in the structure tensor can be readily exploited to adapt the diffusion process to the local image structure. Weickert [45, 46] suggests to perform a principal axis transformation: construct the diffusion tensor $\boldsymbol{D}$ with the same eigenvectors as the structure tensor $\boldsymbol{S}_\varrho$, but change the eigenvalues, depending on the desired functionality of the diffusion: higher values mean simply that the diffusion in the direction of the corresponding eigenvector will be encouraged.

Let us arrange the orthonormal eigenvectors of $\boldsymbol{S}_\varrho$ columnwise into a 2 × 2 matrix $\boldsymbol{X}$; then the characteristic equation reads

$$\boldsymbol{S}_\varrho \boldsymbol{X} = \boldsymbol{X} \ \mathrm{diag}(\mu_1, \mu_2). \tag{2.63}$$

where diag$(\cdot)$ stands for a diagonal matrix.

Let f_1 and f_2 be two continuous functions ($\mathbb{R}^2 \rightarrow [\alpha, 1]$, α is a positive lower bound). Moreover, the function f_1 (which influences the diffusion component parallel to the gradient) has to satisfy the theoretical requirements imposed on the isotropic diffusivity function g (namely its monotone descent) and might be set identical to it, depending only on μ_1; the function f_2 controls the smoothing in the coherence direction.

Let us transform the eigenvalues μ_1, μ_2 of the structure tensor by the functions f_1, f_2 to obtain the eigenvalues φ_1, φ_2 of the diffusion tensor:

$$\begin{aligned} \varphi_1 &= f_1(\mu_1, \mu_2), \\ \varphi_2 &= f_2(\mu_1, \mu_2). \end{aligned} \tag{2.64}$$

Using the new eigenvalues, we arrive at the characteristic equation for the diffusion tensor $\boldsymbol{D}$:

$$\boldsymbol{D}\boldsymbol{X} = \boldsymbol{X}\ \operatorname{diag}(\varphi_1, \varphi_2) \tag{2.65}$$

from which we get directly the formula enabling us to calculate $\boldsymbol{D}$ from the structure tensor's eigenvectors and the adapted eigenvalues (noting that $\boldsymbol{X}\boldsymbol{X}^T = \boldsymbol{I}$):

$$\boldsymbol{D} = \boldsymbol{X}\ \operatorname{diag}(\varphi_1, \varphi_2)\ \boldsymbol{X}^T. \tag{2.66}$$

Clearly, the diffusion tensor $\boldsymbol{D}$ constructed this way satisfies the theoretical requirements of Section 2.3.4.

2.4.3 Choosing the diffusivity functions f_1, f_2

Back to an important detail: what functions $f_{1,2}$ can we choose to convert the structure tensor eigenvalues into the diffusion ones? Weickert in [45] presents two cases:

Edge-enhancing diffusion is intended to overcome the problem of noise remaining near the image edges. The integration parameter ϱ is set to zero, and the diffusion along the image gradient is subjected to the classical diffusivity function of the first eigenvalue μ_1 (which estimates the gradient magnitude), while the diffusion perpendicular to it is encouraged maximally:

$$\begin{aligned} \varphi_1 &= g(\mu_1) \\ \varphi_2 &= 1. \end{aligned} \tag{2.67}$$

Coherence-enhancing diffusion is capable of restoring or enhancing coherent structures (e.g interrupted lines). The integration scale ϱ takes on positive values; the diffusion along coherent structures is preferred (increasing with respect to the coherence measure $(\mu_1 - \mu_2)^2$), a small amount α in other directions remains for theoretical reasons:

$$\begin{aligned} \varphi_1 &= \alpha \\ \varphi_2 &= \begin{cases} \alpha & \text{if } \mu_1 = \mu_2, \\ \alpha + (1-\alpha)\exp\left(\frac{-C}{(\mu_1-\mu_2)^{2m}}\right) & \text{otherwise.} \end{cases} \end{aligned} \tag{2.68}$$

In the following we concentrate on the first case, closer to the general image restoration problem, and suggest to improve it in several ways.

Weickert himself noticed ([45]) that the good noise-filtering behaviour of anisotropic diffusion (esp. with a nonzero integration scale $\varrho > 0$) is paid for by rounding of corners. Additionally, we observed in our experiments that the high value of φ_2 not only removes noise from the edges, but also causes more smoothing in other directions than that of the maximum coherence and leads thus to blurring of important discontinuities. Although this latter objection is linked to the way the diffusion equation is discretized and will be discussed later, there are some simple ways both the drawbacks can be alleviated.

Suggestion 1 : setting the second eigenvalue to a smaller constant value (e.g. $\varphi_2 \in [0.05, 0.25]$) is still sufficient to filter the noise near the image edges. At the same time, both the blurring and the corner rounding are greatly reduced.

Suggestion 2 : alternate the anisotropic NL diffusion and the isotropic NL diffusion in the subsequent iterations. This should also limit the blurring across edges while reducing the overall computational complexity (comparing the mixed isotropic and anisotropic algorithm to the same number of iterations of a purely anisotropic method).

2.5 NL diffusion of vector-valued data

We stated in the introduction that the input data for filtering can be understood as a function $\mathbf{f}\colon \Omega \to \mathbb{R}$. This model covers grey-scale images, range data representing the depths of points in the scene, some medical data etc. In some other cases, several physical quantities are measured at the same location in space: colour (RGB) images fall into this category, together with e.g. multi-echo magnetic resonance medical data, multispectral LANDSAT measurements, and many others. Then, the function $\mathbf{f}$ maps to an m-dimensional space $\mathbb{R}^m$ and the *vector-valued* data $\mathbf{f}$ are represented by a collection of m images, $\mathbf{f} = (\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_m)$.

For vector-valued images, the diffusion equation (2.14) translates into a set of equations

$$\begin{aligned} \partial_t \mathbf{u}_1 &= \operatorname{div}\left(\boldsymbol{D}(\nabla \mathbf{u})\, \nabla \mathbf{u}_1\right) \\ &\vdots \\ \partial_t \mathbf{u}_m &= \operatorname{div}\left(\boldsymbol{D}(\nabla \mathbf{u})\, \nabla \mathbf{u}_m\right) \end{aligned} \tag{2.69}$$

where the solution $\mathbf{u}$ is also composed of m images, $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m)$, with the initial condition $\mathbf{u}_i = \mathbf{f}_i$. Note that the diffusion tensor $\boldsymbol{D}$ depends on the whole vector $\mathbf{u}$ and is identical for all equations in the set (2.69). This has been adopted as a common practice for vector-valued diffusion (see [55, 47, 16]): to avoid inconsistencies between separate channels $\mathbf{u}_i$, the equations (2.69) are coupled and the diffusion of individual images is synchronized through a common set of diffusivities $\boldsymbol{D}$.

The way the common diffusivity is assembled from the vector components depends on the nature of the problem. Going directly to the anisotropic diffusion, Weickert in [47] proposes to construct the common structure tensor $\boldsymbol{S}_\varrho$ for the vector-valued image as a convex combination of the structure tensors of its components:

$$\boldsymbol{S}_\varrho(\nabla \mathbf{u}) = \sum_{i=1}^{m} w_i \, \boldsymbol{S}_\varrho(\nabla \mathbf{u}_i) \tag{2.70}$$

with $\sum w_i = 1$ and $w_i > 0$. In the absence of a priori knowledge on the significance or reliability of individial channels, all weights can be set equal, $w_i = \frac{1}{m}$.

2.6 Numerical methods for NL diffusion

We introduced the nonlinear diffusion, discussed its properties and the meaning of its parameters in the previous sections. This section concentrates on the methods that can be employed to solve, or model, the diffusion equations numerically. Before moving to the discretization issues, we review the continuous formulation here.

We take the filter of Catté *et al.* [10] as a typical representative of a well founded *isotropic* nonlinear diffusion process. With this scheme, the filtered image is found as a solution to the equation

$$\frac{\partial \mathbf{u}}{\partial t} = \operatorname{div}\left(g(|\nabla \mathbf{u}_\sigma|) \, \nabla \mathbf{u}\right) \tag{2.71}$$

with the image $\mathbf{f}$ as the initial state, and the reflecting Neumann boundary conditions,

$$\mathbf{u}(x,0) = \mathbf{f}(x), \qquad \langle \nabla \mathbf{u}, n \rangle = 0 \;\text{ on } \partial\Omega, \tag{2.72}$$

where n denotes the normal to the image boundary $\partial\Omega$. In words, the equation (2.71) expresses the fact that the value of $\mathbf{u}(x,t)$ changes with time according to the flow to and from the neighbourhood of x; this flow depends on the image gradient $\nabla \mathbf{u}$ and its amount is controlled by the scalar diffusivity function g of the smoothed gradient $\nabla \mathbf{u}_\sigma$ (smoothing makes the filter insensitive to noise at scales smaller then σ); no flow passes through the image boundary.

To be suitable for numerical computations with sampled data, the continuous diffusion equation (2.71) has to be discretized. Discretization of partial differential equations represents a vast subject on its own; see e.g. [29] for a brief and general introduction. The issue is even more crucial with the nonlinear diffusion applied to fields of vision or medical imagery where a large quantity of data has to be processed. If many operations can be described in the mathematically sound framework of partial differential equations, computational efficiency may represent an argument against using it; that is why many papers concentrate on numerically efficient implementation of the nonlinear diffusion scheme (e.g. [53, 4, 16]), suggest to employ parallelism [54], reduce the computational cost by using pyramidally subsampled versions of the image [49], or to make use of fast hardware operations [33].

In this section we give an overview of several approaches to PDE discretization. As the discretization of continuous time into discrete steps is considered, explicit and semi-implicit methods have been proposed for the diffusion equation. From the point of view of space discretization, we will distinguish finite elements , finite volume, and finite difference schemes. After this classification, we will concentrate on the finite differences in more detail; we first develop the simplest, explicit (Euler) finite difference discretization of the isotropic diffusion equation, discuss the stability and efficiency issues and move to the absolutely stable, and thus also more efficient additive operator splitting (AOS) scheme of Weickert *et al.* [53]. Having presented numerical methods for the isotropic equations, we continue to the anisotropic NL diffusion, and introduce briefly the Weickert's AOS schemes for the anisotropic NL diffusion [45, 46]. Later in the text (in Section 3.2) we will follow and extend the Weickert's ideas to offer a new directional splitting procedure for the anisotropic AOS scheme.

2.6.1 Time discretization

A vast majority of the numerical schemes for NL diffusion uses first order discretization in time, and approximates $\frac{\partial \mathbf{u}}{\partial t}$ by $\frac{\mathbf{u}^{k+1}-\mathbf{u}^k}{\tau}$ where τ stands for the discrete time step and $\mathbf{u}^k$ is the solution of the diffusion equation at time $t_k = k \cdot \tau$. The methods differ in the way the solutions at times k and $k+1$ are arranged into the equation.

The *explicit* (or Euler forward difference) scheme controls the diffusion by the diffusivities calculated from the previous time step according to

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\tau} = \operatorname{div}\left(g(|\nabla \mathbf{u}_\sigma^k|)\, \nabla \mathbf{u}^k\right). \tag{2.73}$$

The scheme is called explicit because the solution $\mathbf{u}^{k+1}$ is obtained explicitely from $\mathbf{u}^k$ by a simple rearrangement of equation (2.73),

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \tau \cdot \operatorname{div}\left(g(|\nabla \mathbf{u}_\sigma^k|)\, \nabla \mathbf{u}^k\right). \tag{2.74}$$

After space discretization, this scheme can be described by a matrix equation

$$\mathbf{u}^{k+1} = \left(\boldsymbol{I} + \tau \boldsymbol{A}(\mathbf{u}^k)\right)\, \mathbf{u}^k. \tag{2.75}$$

Advantage of the explicite scheme is that very simple operations are performed at every iteration; however, this discretization requires only small time steps in order to be stable [53].

Alternatively, we may employ the *semi-implicit* time discretization [4, 14, 19, 53] of the form

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\tau} = \operatorname{div}\left(g(|\nabla \mathbf{u}_\sigma^k|)\, \nabla \mathbf{u}^{k+1}\right). \tag{2.76}$$

Here the diffusivities g are still computed from the previous time step, but the unknown $\mathbf{u}^{k+1}$ has to be found by solving a system of linear equations

$$\mathbf{u}^k = \left(\boldsymbol{I} + \tau \boldsymbol{A}(\mathbf{u}^k)\right)\, \mathbf{u}^{k+1}. \tag{2.77}$$

The system is stable for all time step sizes, and the sparse control matrix $(\boldsymbol{I} + \tau \boldsymbol{A}(\mathbf{u}^k))$ can be inverted using some iterative techniques, such as Gauss–Seidel or preconditioned conjugate gradient methods.

The *fully implicit* scheme

$$\mathbf{u}^k = \Big(\boldsymbol{I} + \tau \boldsymbol{A}(\mathbf{u}^{k+1})\Big)\ \mathbf{u}^{k+1} \tag{2.78}$$

would involve solution of nonlinear equations without accompanying this difficulty by any special advantages; this approach is not used in practice.

We will present *additive operator splitting* scheme [53] in Section 2.6.4. It may be understood as an attempt to combine the advantages of the explicit and the semi-implicit methods: absolute stability for all time steps, and one iteration consisting of simple and efficient operations.

2.6.2 Discretization in space

From the point of view of space discretization, the numerical methods employed to solve the diffusion equation (2.71) may be classified into finite elements, finite volume and finite differences.

Let us start from the equation

$$\mathbf{u}_t - \operatorname{div}(g\ \nabla\mathbf{u}) = 0, \tag{2.79}$$

multiply it by an arbitrary function $\mathbf{v}$ defined on the same domain Ω as $\mathbf{u}$, and integrate the equation over Ω. We obtain

$$\int_\Omega \mathbf{u}_t\,\mathbf{v}\,\mathrm{d}x - \int_\Omega \operatorname{div}(g\ \nabla\mathbf{u})\,\mathbf{v}\,\mathrm{d}x = 0, \tag{2.80}$$

which can be rewritten (using integration per partes of the second term and Neumann boundary conditions, and discretizing in time) into the formula

$$\int_\Omega \frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\tau}\,\mathbf{v}\,\mathrm{d}x - \int_\Omega g\ \nabla\mathbf{u}\ \nabla\mathbf{v}\,\mathrm{d}x = 0. \tag{2.81}$$

Equation (2.81) is called *weak formulation* of the diffusion equation. The function $\mathbf{u}$ is a weak solution of (2.79) if it satisfies this integral equality for all admissible functions $\mathbf{v}$ (see [14]).

To derive the *finite element* discretization, the domain Ω is partitioned by triangulation and the equation (2.81) is solved by considering the basis functions $\mathbf{v}_h$ which describe all functions $\mathbf{v}$ continuous on Ω and linear on each of the triangles. Then the solution $\mathbf{u}$ (represented by values at the triangle vertices) is also understood as piecewise linear. Computationally, the system boils down to solving systems of linear equations as described in the previous section. Finite elements have been employed for NL diffusion e.g. by Bänsch and Mikula in [4], and by Preußer and Rumpf in [30] (who use bilinear functions on rectangles instead of linear triangular patches in their discretization). Both these papers incorporate mesh coarsening strategies into the algorithms, which reduces the number of unknowns and thus speeds the computations significantly as the diffusion time advances.

Alternatively, we may separate the continuous image domain into small regions and look for a solution which will be constant on each such region. Let us call one such small region V_i, and integrate the diffusion equation (2.79) on it to obtain

$$\int_{V_i} \mathbf{u}_t \,\mathrm{d}x - \int_{V_i} \operatorname{div}(g\,\nabla\mathbf{u})\,\mathrm{d}x = 0. \tag{2.82}$$

Using divergence theorem on the second term we can write

$$\int_{V_i} \mathbf{u}_t \,\mathrm{d}x - \int_{\partial V_i} g\ \langle\nabla\mathbf{u}, n\rangle\ \mathrm{d}x = 0 \tag{2.83}$$

where n is the normal to ∂V_i, the boundary of V_i, and the scalar product $\langle\nabla\mathbf{u}, n\rangle$ represents flux across the boundary. The equation (2.83) represents the *integral form of the diffusion equation*[13]. This integral diffusion formulation leads naturally to the *finite volume* discretization.

Integrating the first term of (2.83) and discretizing in time, we have

$$\int_{V_i} \mathbf{u}_t \,\mathrm{d}x = \frac{\overline{\mathbf{u}}_i^{k+1} - \overline{\mathbf{u}}_i^{k}}{\tau}\,|V_i| \tag{2.84}$$

where $\overline{\mathbf{u}}_i^k$ is the mean value of $\mathbf{u}$ over the volume V_i at time k ($\overline{\mathbf{u}}_i^k$ can be also understood as a representative of $\mathbf{u}$ at V_i and k; the finite volume method considers the solution constant on each V_i), and $|V_i|$ is the measure of V_i.

Let us introduce some more notation: let $\mathcal{N}(i)$ be the set of indeces j such that V_i and V_j are neighbours, i.e. there exists a common interface e_{ij} between V_i and V_j with nonzero measure $|e_{ij}| > 0$. Let for each i there exists a representative point x_i such that for all $j \in \mathcal{N}(i)$, the unit vector normal to e_{ij} can be expressed using these representative points as $n_{ij} = \frac{x_j - x_i}{|x_j - x_i|}$. Let g_{ij} stand for the diffusivity at the edge e_{ij}: with x_{ij} the center point of e_{ij}, we write $g_{ij} = g(|\nabla\mathbf{u}_\sigma(x_{ij})|)$.

With this notation, we can rewrite the flux term of (2.83):

$$\int_{\partial V_i} g\ \langle\nabla\mathbf{u}, n\rangle\ \mathrm{d}x = \sum_{j\in\mathcal{N}(i)} g_{ij}\cdot\frac{\overline{\mathbf{u}}_j - \overline{\mathbf{u}}_i}{|x_j - x_i|}\cdot|e_{ij}|. \tag{2.85}$$

Now let us put (2.84) and (2.85) together and formulate the *finite volume scheme* (semi-implicit in time) for nonlinear diffusion:

$$\left(\frac{|V_i|}{\tau} + \sum_{j\in\mathcal{N}(i)} g_{ij}\cdot\frac{|e_{ij}|}{|x_j - x_i|}\right)\overline{\mathbf{u}}_i^{k+1} - \sum_{j\in\mathcal{N}(i)} g_{ij}\cdot\frac{|e_{ij}|}{|x_j - x_i|}\cdot\overline{\mathbf{u}}_j^{k+1} = \frac{|V_i|}{\tau}\cdot\overline{\mathbf{u}}_i^{k}. \tag{2.86}$$

This scheme was introduced by Mikula and Ramarosy in [19] where the authors also show convergence of the discrete scheme (2.86) to the weak solution of the continuous equation (2.79). It appeared also in the overview [14]; Krivá and Mikula [16]

[13] A small remark: we find the integral form more intuitive and easier to understand than the differential diffusion equation in the divergence form. From (2.83) you can see directly that the value of $\mathbf{u}$ inside the region V_i changes with time exactly with the flux which passes through the region boundary.

combined the finite volumes with mesh adaptivity using quadtrees and applied it to diffusion filtering of colour images.

Finite volumes show the advantages of being conceptually clear, this discretization arises naturally from pixel structure of discrete images. The possibility of grid coarsening allows to speed up the computations and handle large amounts of data (e.g. in three dimensions) efficiently. The numerical computations again restrict, as with all the schemes, to solution of sparse systems of linear equations.

The last scheme of out brief overview, the *finite difference scheme*, is named after the main principle of replacing a partial derivative $\frac{\partial f}{\partial x}$ by a finite difference $\frac{f(x_1)-f(x_2)}{x_1-x_2}$. The finite difference diffusion equation can be derived easily from the finite volume scheme (2.86) for a regular grid. If each pair of representative points has the same distance, $\forall i,\ \forall j \in \mathcal{N}(i) :\ \Delta x = |x_i - x_j|$, we have also $|e_{ij}| = \Delta x$, $|V_i| = \Delta x^2$ (in 2D), and we can rewrite the finite volume scheme into

$$\left(\frac{\Delta x^2}{\tau} + \sum_{j\in\mathcal{N}(i)} g_{ij}\right)\overline{\mathbf{u}}_i^{k+1} - \sum_{j\in\mathcal{N}(i)} g_{ij}\cdot\overline{\mathbf{u}}_j^{k+1} = \frac{\Delta x^2}{\tau}\cdot\overline{\mathbf{u}}_i^k. \tag{2.87}$$

Dropping the bar from the representative values $\overline{\mathbf{u}}_i^k$ and rearranging the equation, we obtain readily the finite difference scheme (semi-implicit in time) as it appeared e.g. in [53]:

$$\mathbf{u}_i^k = \mathbf{u}_i^{k+1} - \tau\cdot\sum_{j\in\mathcal{N}(i)}\frac{g_{ij}}{\Delta x^2}\cdot\left(\mathbf{u}_j^{k+1} - \mathbf{u}_i^{k+1}\right). \tag{2.88}$$

Conceptually, the finite differences are usually treated in a slightly less rigorous way than finite elements or finite volumes, omitting the volumes of the discretization elements or their exact geometry with boundary adjacency, boundary lengths etc. The function $\mathbf{u}$ is represented by values u_i at regularly spaced grid points x_i. The neighbourhood relation is defined in an abstract way from the 4-neighbourhood, 8-neighbourhood, or other topological arrangements. Grid adaptivity is not allowed, the structure is kept regular. However, the scheme is simple to derive and there exist efficient finite difference schemes for nonlinear diffusion, both for the isotropic [53] and for the anisotropic case [45, 23]. We present the explicit and AOS finite difference schemes more thoroughly in the succeeding sections.

2.6.3 Explicit finite difference discretization scheme

Let us start for simplicity with a one-dimensional, isotropic case, for which the diffusion equation (2.57) simplifies to

$$\partial_t\mathbf{u} = \partial_x\left(g(|\partial_x\mathbf{u}_\sigma|)\,\partial_x\mathbf{u}\right). \tag{2.89}$$

For discrete data u_i^k (approximating $\mathbf{u}$ at position $x_i = i\cdot\Delta x$ and time instant $t_k = k\cdot\tau$, with τ the discretization time step and Δx the spatial grid size), replacing the derivatives by finite differences, the equation (2.89) becomes

$$\frac{u_i^{k+1} - u_i^k}{\tau} = \sum_{j\in\mathcal{N}(i)}\frac{g_{ij}^k}{\Delta x^2}(u_j^k - u_i^k) \tag{2.90}$$

where $\mathcal{N}(i)$ is the set of the neighbours of pixel i and g_{ij}^k is the diffusivity belonging to the connection between pixels i and j at time t_k.

The equation (2.90) is called the explicit discretization scheme of (2.89); the solution at time $t+1$ can be obtained from the solution at time t explicitly by the following formula which summarizes the equation (2.90) using a matrix notation:

$$\mathbf{u}^{k+1} = \Big(\boldsymbol{I} + \tau \boldsymbol{A}(\mathbf{u}^k)\Big)\, \mathbf{u}^k, \tag{2.91}$$

where τ is a discrete time step, $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{A}(\mathbf{u}^k)$ contains the diffusivity information:

$$a_{i,j} = \begin{cases} \dfrac{g_{ij}^k}{\Delta x^2} & \text{for } j \in \mathcal{N}(i) \\ -\displaystyle\sum_{n \in \mathcal{N}(i)} \dfrac{g_{in}^k}{\Delta x^2} & \text{for } j = i \\ 0 & \text{otherwise} \end{cases} \tag{2.92}$$

(note that only the elements $a_{i,j}$ of $\boldsymbol{A}$ for which either $j \in \mathcal{N}(i)$ of $i = j$ are nonzero; in 1D $\boldsymbol{A}$ is tridiagonal). For two-dimensional data $\mathbf{u}$ another term appears:

$$\mathbf{u}^{k+1} = \Big(I + \tau \boldsymbol{A}_x(\mathbf{u}^k) + \tau \boldsymbol{A}_y(\mathbf{u}^k)\Big)\, \mathbf{u}^k \tag{2.93}$$

and $\boldsymbol{A}_x(\mathbf{u}^k)$ and $\boldsymbol{A}_y(\mathbf{u}^k)$ are matrices containing information about the diffusivities between individual pixels in the directions of axes x and y, respectively. There are only two differences: the diffusivities g_{ij} to fill the matrices $\boldsymbol{A}_l$ are computed as a function of a two-dimensional gradient size $g(|\nabla \mathbf{u}_\sigma|) = g\big(\sqrt{|\partial_x \mathbf{u}_\sigma|^2 + |\partial_y \mathbf{u}_\sigma|^2}\,\big)$, and the pixels of $\mathbf{u}$ must be arranged into a single column vector to allow the matrix multiplication.

The explicit (or Euler) discretization scheme used in this section is the most straightforward one but requires a small time step τ in order to be stable. Weickert shows in [53] that the stability condition (assuming $\Delta x = 1$ and $\forall s : \; g(s) \leq 1$) is

$$\tau < \frac{1}{2N} \tag{2.94}$$

with N being the number of dimensions of the data. This is a severe limitation, implying that more iterations are needed to reach a fixed stopping time T, and that poor efficiency may prevent nonlinear diffusion from being applied in practical situations. Slightly more complex, but absolutely stable, and thus more efficient methods are discussed in the sections below.

2.6.4 AOS finite difference scheme for nonlinear diffusion

In [53], Weickert *et al.* suggest to replace the explicit method by the *additive operator splitting (AOS)* scheme which separates and discretizes the diffusion equation (2.57) by

$$\mathbf{u}^{k+1} = \frac{1}{m} \sum_{l=1}^{m} (\boldsymbol{I} - m\tau \boldsymbol{A}_l(\mathbf{u}^k))^{-1} \mathbf{u}^k \tag{2.95}$$

with l the direction index, $l = 1,\ldots,m$ (for isotropic diffusion, m is set equal to the dimensionality of the data; for anisotropic diffusion, larger values of m will be necessary).

Each of the summands in (2.95) represents a one-dimensional diffusion process along the direction l. The compound diffusion iteration is obtained as an average of these one-dimensional processes, regardless of the dimensionality of the input data.

Both the AOS and the explicit schemes are of the same approximation order (first order in τ, second order in the grid size Δx) to the continuous diffusion equation (2.71); this can be easily checked if you compare the equations (2.93) and (2.95), and use the equality $(\boldsymbol{I} - \alpha \boldsymbol{A})^{-1} = \boldsymbol{I} + \alpha \boldsymbol{A} + (\alpha \boldsymbol{A})^2 + \cdots$. In this sense, the two discretizations are equivalent.

The matrices $\boldsymbol{A}_l$ which store the diffusivity information for the diffusion direction l are formed in the same way as with the explicit algorithm. The matrices $\boldsymbol{I} - m\tau \boldsymbol{A}_l(\mathbf{u}^k)$ can be made tridiagonal and diagonally dominant by a simple rearrangement of the pixels, and then inverted efficiently by Thomas algorithm [53]. This way, one iteration of the AOS scheme requires only about twice the computational effort needed for one iteration of the explicit scheme. What we gain for this price is absolute stability: the AOS scheme is stable and creates a discrete scale space for any choice of the discretization step τ. The freedom to select a larger τ means that fewer iterations are needed to reach a fixed stopping time T, and the algorithm becomes faster. Although a large τ also weakens the filtering effect and the solution may become less precise an approximation to the ideal continuous solution, Weickert *et al.* [53] report that for typical precision requirements of 2%[14], the AOS scheme is at least 11 times faster than any stable explicit scheme.

The AOS scheme can be extended to any number of dimensions. Also, as each direction and each line in that direction can be processed independently from other lines/directions, a parallel implementation is straightforward [54].

2.6.5 AOS finite difference scheme for anisotropic diffusion

Moving to the anisotropic diffusion for which the diffusivity need not be equal in all directions, we again want to approximate the continuous process by a discrete algorithm. Again, the AOS scheme will separate the 2D diffusion into several one-dimensional diffusion processes along chosen directions. However, the anisotropic diffusion will need more directions than the isotropic filter did. There exists a direct relation between the number of one-dimensional processes and the achievable anisotropy of the compound diffusion filter. For sake of simplicity and computational efficiency, we restrict the approximation to four directions defined by the boundary pixels of a 3 × 3 window. In this situation, the AOS discretization is

[14] The precision was computed with respect to an explicit scheme with a very small time step $\tau = 0.1$ which was proclaimed the ground truth.

computed according to

$$\mathbf{u}_l^{k+1} = \frac{1}{4} \sum_{l=-1}^{2} (\boldsymbol{I} - 4\tau \boldsymbol{A}_l(\mathbf{u}^k))^{-1} \mathbf{u}^k. \tag{2.96}$$

The main difficulty is to split the 2D diffusion tensor $\boldsymbol{D}$ correctly into one-dimensional diffusivities to fill the matrices $\boldsymbol{A}_l$.

In [45], Weickert gives a constructive proof of the following theorem: given $\boldsymbol{D}$, a symmetric positive definite matrix with a spectral condition number κ, there exists some $n(\kappa) \in \mathbb{N}$ such that $\operatorname{div}(\boldsymbol{D} \cdot \nabla \mathbf{u})$ reveals a second-order nonnegative forward difference discretization on a $(2n+1) \times (2n+1)$ window.

The boundary pixels of a $(2n+1) \times (2n+1)$ window define $4n$ principal orientations $\beta_i \in (-\frac{\pi}{2}, \frac{\pi}{2}]$, $i = -2n+1, \ldots, 2n$. The theorem says that it is possible to separate the continuous process into $4n$ one-dimensional processes along these orientations. Moreover, Weickert showed that only three of these orientations are actually needed to guarantee the positive discretization at any single location, so that we end up with the approximation

$$\operatorname{div}(\boldsymbol{D} \cdot \nabla \mathbf{u}) = \partial_{e_{\beta_0}}(\alpha_0\, \partial_{e_{\beta_0}} \mathbf{u}) + \partial_{e_{\beta_k}}(\alpha_k\, \partial_{e_{\beta_k}} \mathbf{u}) + \partial_{e_{\beta_{2n}}}(\alpha_{2n}\, \partial_{e_{\beta_{2n}}} \mathbf{u}) \tag{2.97}$$

where $e_{\beta_i} = (\cos\beta_i, \sin\beta_i)^T$, and $\alpha_0, \alpha_k, \alpha_{2n}$ are the nonnegative directional diffusivities along the orientations $\beta_0, \beta_k, \beta_{2n}$.

For the particular case of 2D data and a 3×3 window (implying four principal orientations[15] $\beta_{-1} = -\frac{\pi}{4}$, $\beta_0 = 0$, $\beta_1 = \frac{\pi}{4}$, $\beta_2 = \frac{\pi}{2}$) with the diffusion tensor $\boldsymbol{D} = \left(\begin{smallmatrix} a & b \\ b & c \end{smallmatrix}\right)$, its condition number bounded by $\kappa_{max} = 3 + 2\sqrt{2}$, Weickert proposes the following directional diffusivities:

$$\alpha_{-1} = \frac{|b| - b}{2} \qquad \alpha_0 = a - |b| \tag{2.98}$$

$$\alpha_1 = \frac{|b| + b}{2} \qquad \alpha_2 = c - |b|. \tag{2.99}$$

The directional diffusivities are assembled into matrices $\boldsymbol{A}_l$, and the AOS discretization of the 2D continuous anisotropic diffusion process is computed using the equation (2.96). This directional splitting leads to an algorithm which reveals (discrete versions of) all theoretical properties of the continuous diffusion filter presented in Section 2.3.4. As for practical properties, some problems arise with the rotational symmetry of the discrete filter; we return to this point in Section 3.2.

In a private communication, Joachim Weickert suggested that any diffusivity splitting method can only possess two of these three desirable properties:

1. positivity (max-min principle);

2. anisotropy (strong directionality without limits on the condition number of $\boldsymbol{D}$);

[15] We assume equal grid size $\Delta x = 1$ in both dimensions.

3. consistency with the continuous equation (the splitting should converge to the continuous equation as the discretization steps approach zero, $\tau \to 0$, $\Delta x \to 0$).

Weickert and Scharr renounced the positivity to obtain rotational symmetry and good directionality (with less blurring in other than coherence directions) in [52]. The disadvantage of that choice is that the maximum-minimum principle is lost, some oscillations may appear, and the method is not suitable for noise filtering.

3 Three diffusion fragments

We present here the contributions that the author has made to the general (isotropic and anisotropic) nonlinear diffusion as applied to image filtering. The first part (Section 3.1) reopens an important question: at what time T should we stop the diffusion process in order to obtain the best filtering result? The second one (sec. 3.2) studies one detail of the anisotropic diffusion: what is the correct way to split the continuous anisotropic diffusion process into four one-dimensional processes as needed for the numerical additive operator splitting scheme. The two topics have been published in [22] and [23], respectively. The last section in this chapter then summarizes all the ideas mentioned above into a complete and autonomous image filtering algorithm.

3.1 Optimal stopping time selection for NL diffusion

Section 2.3.5 discussed the role of stopping time T for nonlinear diffusion. We also mentioned the previously suggested strategies to select a good T, but were not satisfied with any of them. We offer a new stopping-time selection method in this section.

We work with the following model (see Figure 3.1): let $\widetilde{\mathbf{f}}$ be the ideal, noise-free (discrete) image; this image is observed by some imprecise measurement device to obtain an image $\mathbf{f}$. We assume that some noise n is added to the signal during the observation so that

$$\mathbf{f} = \widetilde{\mathbf{f}} + n. \tag{3.1}$$

Furthermore, we assume that the noise n is uncorrelated with the signal $\widetilde{\mathbf{f}}$, and that the noise has zero mean value, $E(n) = 0$.[1]

[1] Let us review the statistical definitions used in this section (see e.g. Papoulis [27]). For the statistical computations on images, we treat the pixels of an image as independent observations of a random variable.
The *mean* or *expectation* of a vector x is $\overline{x} = E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$.
We define the *variance* of a signal x as $\operatorname{var}(x) = E\big[(x-\overline{x})^2\big]$.
The *covariance* of two vectors x, y is given by $\operatorname{cov}(x,y) = E\left[(x-\overline{x})\cdot(y-\overline{y})\right].$

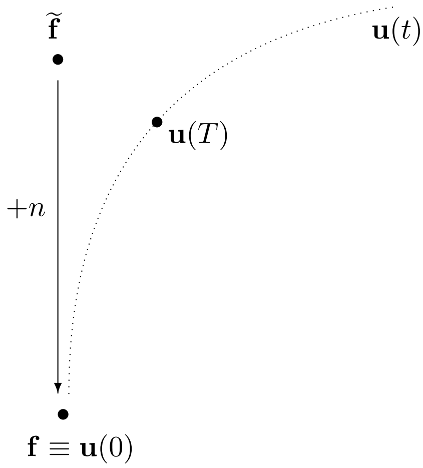


Figure 3.1: Model of the time-selection problem for the diffusion filtering. We want to select the filtered image $\mathbf{u}(T)$ which is as close as possible to the ideal signal $\widetilde{\mathbf{f}}$.

The diffusion filtering starts with the noisy image as its initial condition, $\mathbf{u}(0) = \mathbf{f}$, and the diffusion evolves along some trajectory $\mathbf{u}(t)$. This trajectory depends on the diffusion parameters and on the input image; the optimistic assumption is that the noise will be removed from the data before important features of the signal commence to deteriorate significantly, so that the diffusion leads us somewhere 'close' to the ideal data. This should be the case if the signal adheres to the piecewise constant model inherent in the diffusion equation.

The task of the stopping time selection can be formulated as follows: select that point $\mathbf{u}(T)$ of the diffusion evolution which is nearest to the ideal signal $\widetilde{\mathbf{f}}$. Obviously, the ideal signal is normally not available; the optimal stopping time T can only be estimated by some criteria, and the distance[2] between the ideal and the filtered data serves only in the experiments to evaluate the performance of the estimation procedure.

We introduced the previous approaches to the time selection problem in Section 2.3.5, and were rather critical about their applicability. Even the most promising of them, the Weickert's formula (2.56), based on the knowledge of the noise variance and measuring somehow the distance of the filtered image from its noisy initialization, does not usually yield the optimal stopping time. Let us study in more detail why the problems occur.

The equality (2.54) and hence the equation (2.55) are valid only if the signal and the noise are uncorrelated. This assumption is valid for $\widetilde{\mathbf{f}}$ and n, but not necessarily for the filtered signal $\mathbf{u}(T)$ and the difference $\mathbf{u}(0) - \mathbf{u}(T)$; the latter

The normalized form of the covariance is called the *correlation coefficient*, $\mathrm{corr}(x, y) = \frac{\mathrm{cov}(x,y)}{\sqrt{\mathrm{var}(x)\cdot\mathrm{var}(y)}}$.

[2] In the experiments below, we measure the distance of two images by the *mean absolute deviation*, $\mathrm{MAD}(x - y) = E\left(|x - y - E(x - y)|\right)$. If the two random variables x, y share the same mean value, we have $E(x - y) = 0$ and the MAD distance simplifies to $\mathrm{MAD}(x - y) = E(|x - y|)$. We can use this latter formula as neither the noise nor the diffusion process change the average grey value of the data, $E\big(\widetilde{\mathbf{f}} - \mathbf{u}(t)\big) = 0$.

is needed for the equation (2.56) to be justified. In other words (if we substitute mentally the filtered function $\mathbf{u}(T)$ for $\widetilde{\mathbf{f}}$, the difference $n_u \equiv \mathbf{u}(0) - \mathbf{u}(T)$ for the noise n, and $\mathbf{u}(0)$ for $\mathbf{f}$ in (2.54) and (2.55)), the formula (2.56) is useful only if the random variables $\mathbf{u}(T)$ and $(\mathbf{u}(0) - \mathbf{u}(T))$ are uncorrelated.

The influence of the covariance on the equations is even better visualised if we compute the difference of the general form $\operatorname{var}(n_u) = \operatorname{var}(\mathbf{u}(0) - \mathbf{u}(t))$ and the 'uncorrelated' result[3] $\operatorname{var}(n_u) \approx \operatorname{var}(\mathbf{u}(0)) - \operatorname{var}(\mathbf{u}(t))$:

$$\begin{aligned}
&\operatorname{var}\big(\mathbf{u}(0) - \mathbf{u}(t)\big) \;-\; \Big(\operatorname{var}\big(\mathbf{u}(0)\big) - \operatorname{var}\big(\mathbf{u}(t)\big)\Big) = \\
&\quad = \operatorname{var}\big(\mathbf{u}(0)\big) + \operatorname{var}\big(\mathbf{u}(t)\big) - 2\cdot\operatorname{cov}\big(\mathbf{u}(0), \mathbf{u}(t)\big) - \Big(\operatorname{var}\big(\mathbf{u}(0)\big) - \operatorname{var}\big(\mathbf{u}(t)\big)\Big) \\
&\quad = 2\cdot\Big(\operatorname{var}\big(\mathbf{u}(t)\big) - \operatorname{cov}\big(\mathbf{u}(0), \mathbf{u}(t)\big)\Big) \\
&\quad = 2\cdot\bigg(E\Big[\Big(\mathbf{u}(t) - \overline{\mathbf{u}(t)}\Big)^2\Big] - E\Big[\Big(\mathbf{u}(0) - \overline{\mathbf{u}(0)}\Big)\cdot\Big(\mathbf{u}(t) - \overline{\mathbf{u}(t)}\Big)\Big]\bigg) \\
&\quad = 2\cdot E\Big[\Big(\big(\mathbf{u}(t) - \overline{\mathbf{u}(t)}\big) - \big(\mathbf{u}(0) - \overline{\mathbf{u}(0)}\big)\Big)\cdot\Big(\mathbf{u}(t) - \overline{\mathbf{u}(t)}\Big)\Big] \\
&\quad = 2\cdot E\Big[\Big(\mathbf{u}(t) - \mathbf{u}(0) - \overline{(\mathbf{u}(t) - \mathbf{u}(0))}\Big)\cdot\Big(\mathbf{u}(t) - \overline{\mathbf{u}(t)}\Big)\Big] \\
&\quad = -2\cdot\operatorname{cov}\big(\mathbf{u}(0) - \mathbf{u}(t), \mathbf{u}(t)\big).
\end{aligned} \tag{3.2}$$

Inspired by these computations, we arrive to the following idea: if the unknown noise n is uncorrelated with the unknown signal $\widetilde{\mathbf{f}}$, wouldn't it be reasonable to minimize the covariance of the 'noise' $(\mathbf{u}(0) - \mathbf{u}(t))$ with the 'signal' $\mathbf{u}(t)$, or – better – employ its normalized form, the correlation coefficient

$$\operatorname{corr}\big(\mathbf{u}(0) - \mathbf{u}(t), \mathbf{u}(t)\big) = \frac{\operatorname{cov}\big(\mathbf{u}(0) - \mathbf{u}(t), \mathbf{u}(t)\big)}{\sqrt{\operatorname{var}\big(\mathbf{u}(0) - \mathbf{u}(t)\big)\cdot\operatorname{var}\big(\mathbf{u}(t)\big)}} \tag{3.3}$$

and choose the stopping time T so that the expression (3.3) is as small as possible? This way, instead of determining the stopping time so that $(\mathbf{u}(0) - \mathbf{u}(T))$ satisfies a quantitative property and its variance is equal to the known variance of the noise n, we try to enforce a qualitative feature: if the ideal $\widetilde{\mathbf{f}}$ and n were uncorrelated, we require that their artificial substitutes $\mathbf{u}(T)$ and $(\mathbf{u}(0) - \mathbf{u}(T))$ reveal the same property, to the extent possible, and select

$$T = \arg\min_t \operatorname{corr}\big(\mathbf{u}(0) - \mathbf{u}(t), \mathbf{u}(t)\big). \tag{3.4}$$

We will call formula (3.4) *decorrelation criterion*; let us test and validate its performance experimentally.

We took the cymbidium image shown in Figure 5.5, added various levels of Gaussian noise to it, filtered by nonlinear diffusion, and observed how the signal–noise correlation measured by equation (3.3) develops with time. A typical example

[3] Meaning again that the random variables $\mathbf{u}(T)$ and $(\mathbf{u}(0) - \mathbf{u}(T))$ should be uncorrelated.

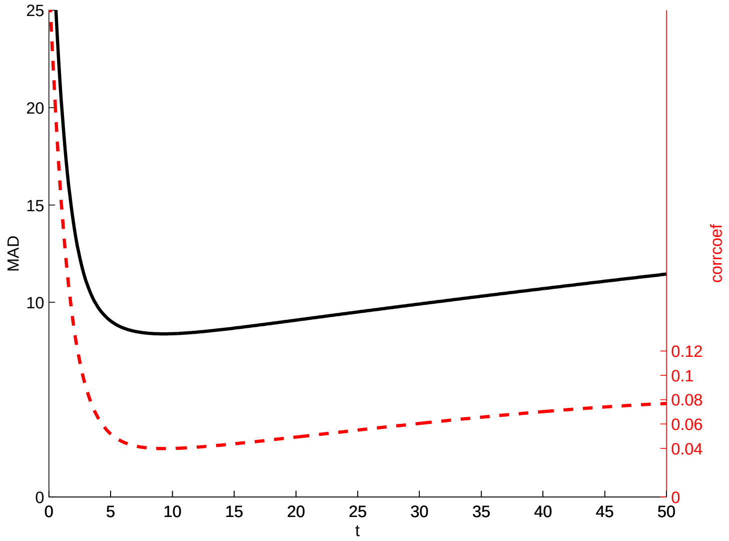


Figure 3.2: The distance $\mathrm{MAD}(\mathbf{u}(t) - \widetilde{\mathbf{f}})$ (solid line) and the correlation coefficient $\mathrm{corr}\big(\mathbf{u}(0) - \mathbf{u}(t), \mathbf{u}(t)\big)$ (dashed line) developing with the diffusion time. The input file 'p5n.pgm' from the cymbidium experiment in Section 5.1.2 was filtered with the diffusion parameters $\sigma = 1$, $\tau = 0.1$, and the Black-Sapiro's $\lambda = 192.8$.

is drawn in Figure 3.2: you can observe that the plot of the MAD criterion measuring the actual filtering quality coincides very well with the graph of the correlation coefficient $\mathrm{corr}\big(\mathbf{u}(0) - \mathbf{u}(t), \mathbf{u}(t)\big)$.

A more thorough study of the performance of the stopping time selection criteria (measured again on the cymbidium data of Section 5.1.2) is seen in figures 3.3 and 3.4. The former compares three stopping times: the optimal T_{opt} is the time instant for which the filtered image $\mathbf{u}(t)$ is closest to the noise-free $\widetilde{\mathbf{f}}$ in the MAD distance; obviously, T_{opt} can be found only in the artificial experimental setting, the noise-free $\widetilde{\mathbf{f}}$ is normally not available. The second stopping time T_{SNR} is determined using the criterion (2.56) (which requires the knowledge of the noise variance or SNR; exact SNR values were supplied to the procedure in these artificial settings). The stopping time T_{corr} minimizing the signal-noise correlation is computed using equation (3.4). All alternative stopping times are computed for a series of input images with varied amount of noise present. While the SNR method easily underestimates or overestimates the optimal stopping time (depending on the amount of noise in the input data), the correlation minimization leads to near-optimal results for all noise levels and all time step sizes.

The actually obtained quality measure $\mathrm{MAD}(\mathbf{u}(T) - \widetilde{\mathbf{f}})$ is shown in Fig. 3.4. The graphs are plotted for the iteration step $\tau = 0.5$. You can see that for all noise levels the correlation-estimated time leads to filtering results very close to the optimal values obtainable by the nonlinear diffusion.

Let us return for a moment to Figure 3.2. At the beginning of the diffusion filtering, the correlation coefficient declines fast until it reaches its minimum. If for some data the graph behaves differently, it may serve as a hint on some problems.

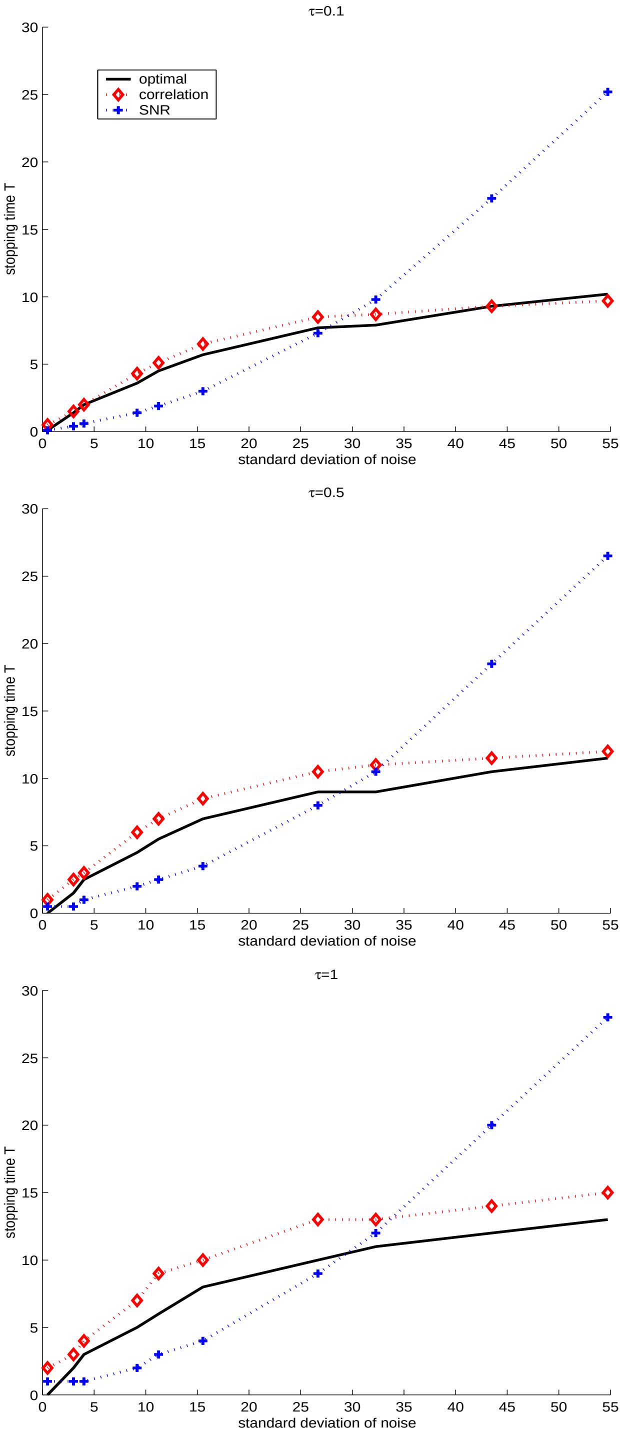


Figure 3.3: The stopping time T_{SNR} determined by the SNR method (dotted with crosses), and T_{corr} obtained through the covariance minimization (dotted with diamonds) compared to the optimal stopping time T_{opt} (solid line). The graphs are plotted against the standard deviation of noise in the input image; the three figures represent the same measurements for different iteration time-step sizes (top to bottom): $\tau = 0.1$, $\tau = 0.5$, $\tau = 1$.

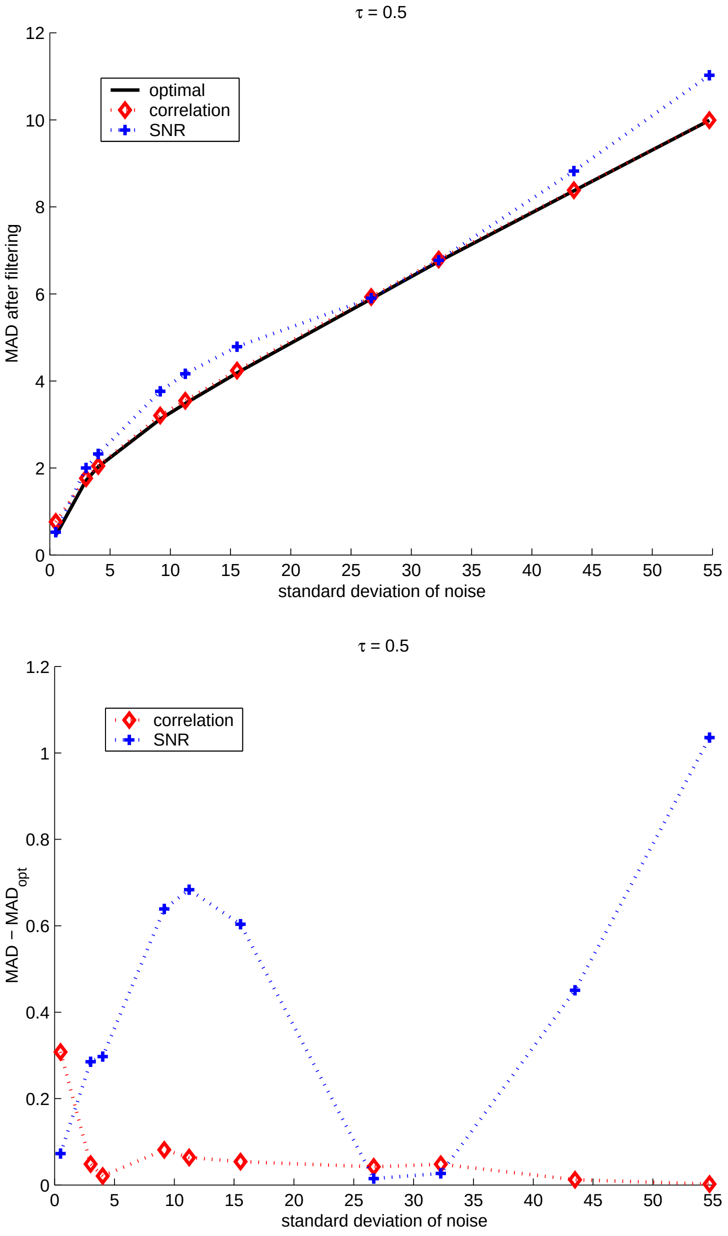


Figure 3.4: Top: the MAD distance of the filtered data from the ideal noise-free image, $\mathrm{MAD}(\mathbf{u}(T) - \widetilde{\mathbf{f}})$, using the SNR and the correlation-minimization time selection strategies. Bottom: the difference between the estimated result and the optimal one, $\mathrm{MAD}(\mathbf{u}(T) - \mathbf{u}(T_{\mathrm{opt}}))$.

As an example, we observed that if there is only a small amount of noise in the input image, the correlation $\operatorname{corr}\big(\mathbf{u}(0)-\mathbf{u}(t),\mathbf{u}(t)\big)$ might grow from the first iterations. In such a case, the iteration time step τ has to be decreased adaptively and the diffusion restarted from time $t=0$ until the correlation plot exhibits a clear minimum.

We conclude this section on the time-selection strategies by stressing that not only the correlation-minimization outperforms all other methods we know of: it is also more general, being based only on the assumption that the noise n is uncorrelated with the original data $\widetilde{\mathbf{f}}$. We do not need any knowledge on the variance of the noise or the signal-to-noise ratio, neither do we need to tune any parameters on 'typical' images. On the other hand, we should not forget that the decorrelation criterion can only *estimate* a good T; if some other information (e.g. the expected noise variance) is available, two or more time-selection strategies can be combined to increase reliability of the estimates.

Some images filtered using the correlation stopping time selection method are to be seen in Section 5.1.

3.2 Consistent positive splitting of anisotropic NL diffusion

In this section, we follow up the topic of Section 2.6.5 (numerical methods for anisotropic diffusion) with a more detailed study of the consistent directional splitting of anisotropic diffusion on a 3 × 3 window. The splitting is needed to obtain discrete numerical algorithms which approximate the continuous 2D anisotropic diffusion process by several (four in our case) one-dimensional procedures acting along chosen directions.

Let us start from the diffusion tensor $\boldsymbol{D}=\left(\begin{smallmatrix}a & b\\ b & c\end{smallmatrix}\right)$ formed as in Section 2.4. We want to approximate the (continuous) 2D diffusion by a diffusion composed of 1D processes acting along four directions $\beta_{-1}=-\frac{\pi}{4}$, $\beta_0=0$, $\beta_1=\frac{\pi}{4}$, $\beta_2=\frac{\pi}{2}$; we will call the approximation *consistent* if

$$\sum_{k=-1}^{2}\partial_{e_{\beta_k}}(\alpha_k\cdot\partial_{e_{\beta_k}}\mathbf{u})=\operatorname{div}(\boldsymbol{D}\cdot\nabla\mathbf{u}) \tag{3.5}$$

where α_k is the diffusivity along the direction β_k.

Let us expand the left hand side of equation (3.5). With $e_{\beta_k}=(\cos\beta_i,\sin\beta_i)^T$ and using the notation $u_x=\frac{\partial\mathbf{u}}{\partial x}$ (similarly for u_y, u_{xx}, etc.), we have

$$\partial_{e_{\beta_k}}=\langle e_{\beta_k},\nabla\mathbf{u}\rangle=u_x\,\cos\beta_k+u_y\,\sin\beta_k \tag{3.6}$$

from which

$$\partial_{e_{\beta_k}}(\alpha_k\cdot\partial_{e_{\beta_k}}\mathbf{u})=\alpha_k\cdot\big(u_{xx}\,\cos^2\beta_k+2u_{xy}\,\sin\beta_k\,\cos\beta_k+u_{yy}\,\sin^2\beta_k\big)\,. \tag{3.7}$$

The right hand side of (3.5) yields

$$\operatorname{div}(\boldsymbol{D}\cdot\nabla\mathbf{u})=\left\langle\begin{pmatrix}\partial_x\\ \partial_y\end{pmatrix},\begin{pmatrix}a\,u_x+b\,u_y\\ b\,u_x+c\,u_y\end{pmatrix}\right\rangle \qquad =a\,u_{xx}+2b\,u_{xy}+c\,u_{yy}. \tag{3.8}$$

If we evaluate the trigonometric functions for the angles β_k in (3.7), sum up the contributions for all k, and assemble the elements corresponding to a given partial derivative of $\mathbf{u}$, we obtain the following set of linear equations:

$$\begin{aligned}
&\text{for } u_{xx}: && \frac{1}{2}\alpha_{-1} + \alpha_0 + \frac{1}{2}\alpha_1 && = a && (3.9)\\
&\text{for } u_{xy}: && -\alpha_{-1} \qquad + \alpha_1 && = 2\,b && (3.10)\\
&\text{for } u_{yy}: && \frac{1}{2}\alpha_{-1} \qquad + \frac{1}{2}\alpha_1 + \alpha_2 && = c. && (3.11)
\end{aligned}$$

We want to solve this set of equations for the unknown α_k with the restriction $\alpha_k \geq 0$, $\forall k$ (the splitting should be positive).

Let us take the second equation of the system, (3.10), and add another equation to it, formed for a parameter $p = \frac{\alpha_{-1}+\alpha_1}{2}$:

$$-\alpha_{-1} + \alpha_1 = 2\,b \tag{3.12}$$

$$\alpha_{-1} + \alpha_1 = 2\,p. \tag{3.13}$$

Summing and subtracting of the two equations lead to

$$\alpha_1 = b + p \quad \geq 0 \tag{3.14}$$

$$\alpha_{-1} = p - b \quad \geq 0 \tag{3.15}$$

from which we obtain the first solvability condition, $p \geq |b|$.

Using the parameter p, the solutions of (3.9)–(3.11) may be expressed as

$$\alpha_0 = a - p \qquad\qquad \alpha_2 = c - p \tag{3.16}$$

$$\alpha_{-1} = p - b \qquad\qquad \alpha_1 = p + b. \tag{3.17}$$

As we require $\alpha_k \geq 0$, the equation (3.16) provides directly the upper bound on our splitting parameter: $p \leq \min(a, c)$. To summarize, the consistent splitting (3.16)–(3.17) remains positive if we select

$$p \in \big[|b|, \min(a, c)\big]. \tag{3.18}$$

The conditions ensuring that this interval is nonempty can be expressed in terms of the condition number of the diffusion tensor $\boldsymbol{D}$. This has been done by Weickert [45] and we have seen it in Section 2.6.5: the consistent positive splitting on a 3 × 3 window is possible if the condition number of $\boldsymbol{D}$ is smaller of equal to $\kappa_{max} = 3 + 2\sqrt{2}$.

Let us now return to an important point: which value from the admissible interval should be chosen for the parameter p? In the following we offer three possibilities:

Splitting 1: $p = |b|$

Splitting 2: $p = \min(a, c)$

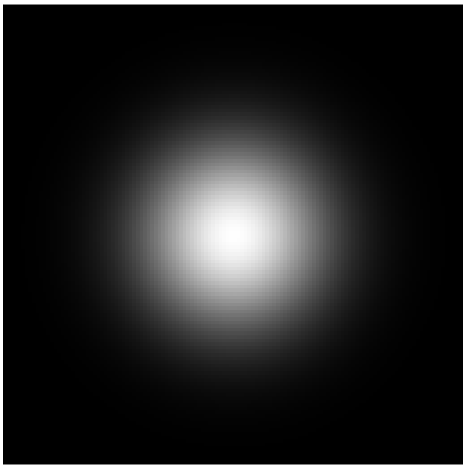


Figure 3.5: Gaussian hill, the input data to test the rotational symmetry of the methods.

Splitting 3: $p = \frac{|b| + \min(a,c)}{2}$.

The first two suggestions take on the value of either of the limit cases; splitting 1 is clearly equivalent to Weickert's positive splitting (2.98)–(2.99). The third splitting represents a compromise, an average of the two limits of the admissible interval (3.18).

The three splitting alternatives are tested for rotational symmetry in Figure 3.6, showing a diffused Gaussian hill. You can observe that although all the three methods share the same property of consistency with the continuous formulation, splitting 1 and 2 does not transfer the continuous rotational symmetry well into the discrete situation. The artefacts become severe as the iteration time step τ increases. On the other hand, the compromising splitting 3 restricts the artefacts considerably, both for a large $\tau = 10$ and for a small $\tau = 1$ (the latter situation is stressed in Figure 3.7 depicting the difference between the diffusion result and its copy rotated by 45 degrees). Figure 3.7d illustrates that if the diffusivity parameter λ is (too) small, the interval (3.18) becomes narrow on a large part of the image, and the three splitting alternatives perform equivalently in preferring some diffusion directions to others, which may create some irregular patterns in the data.

The noise-filtering capabilities of splitting 1 and splitting 3, tested on the data from Figure 5.1, are compared in Figure 3.8. You can see that some cloth-like artefacts remain in the filtered image with fewer iterations of splitting 1, and the output data for $\tau = 1$ and $\tau = 5$ differ significantly. In contrast with that, all the results of splitting 3 resemble, although – as usual for the AOS scheme – the filtering effect weakens as the discretization time step increases. These results prove that the differences between the splitting methods might be negligible if a small time step τ is employed. However, if you ask for a more efficient algorithm and wish to spend fewer iterations of the filtering procedure, thus needing a larger τ, splitting 3 becomes clearly superior.

To summarize this section: we have analysed the possibilities for consistent positive directional splitting of anisotropic diffusion on a 3×3 window. We have shown that such a splitting exists if the interval $[b, min(a, c)]$, formed from the

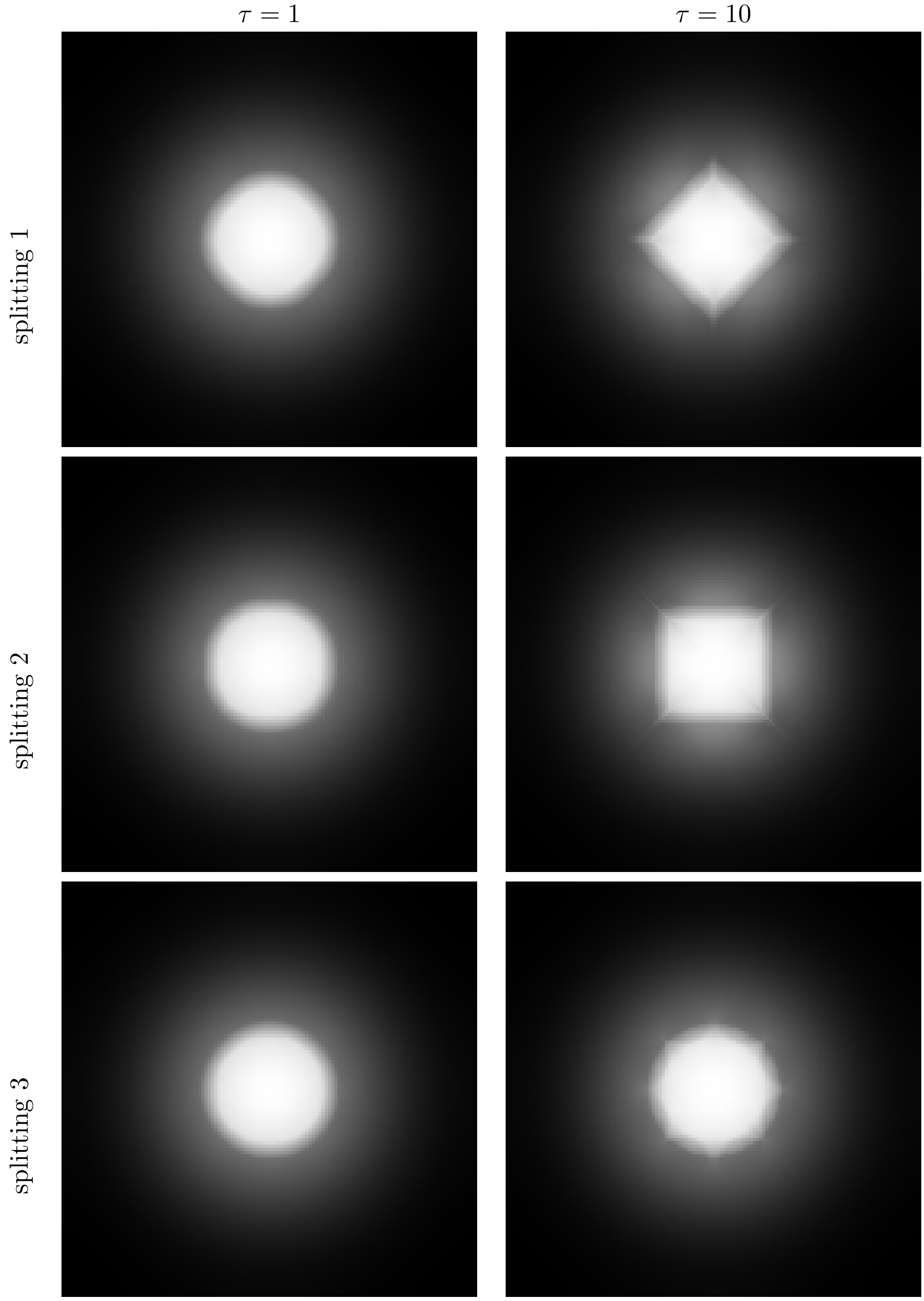


Figure 3.6: A rotationally symmetric Gaussian hill from Figure 3.5 filtered by the three directional splitting methods for the anisotropic NL diffusion AOS scheme of Section 2.6.5. Left column contains the results for the discretization time step $\tau = 1$, the right for $\tau = 10$. The other diffusion parameters were $\sigma = 1$, $T = 200$, $\varphi_2 = 1$; the parameter λ was computed in each diffusion step as the 95$^{\text{th}}$ percentile of image gradients.

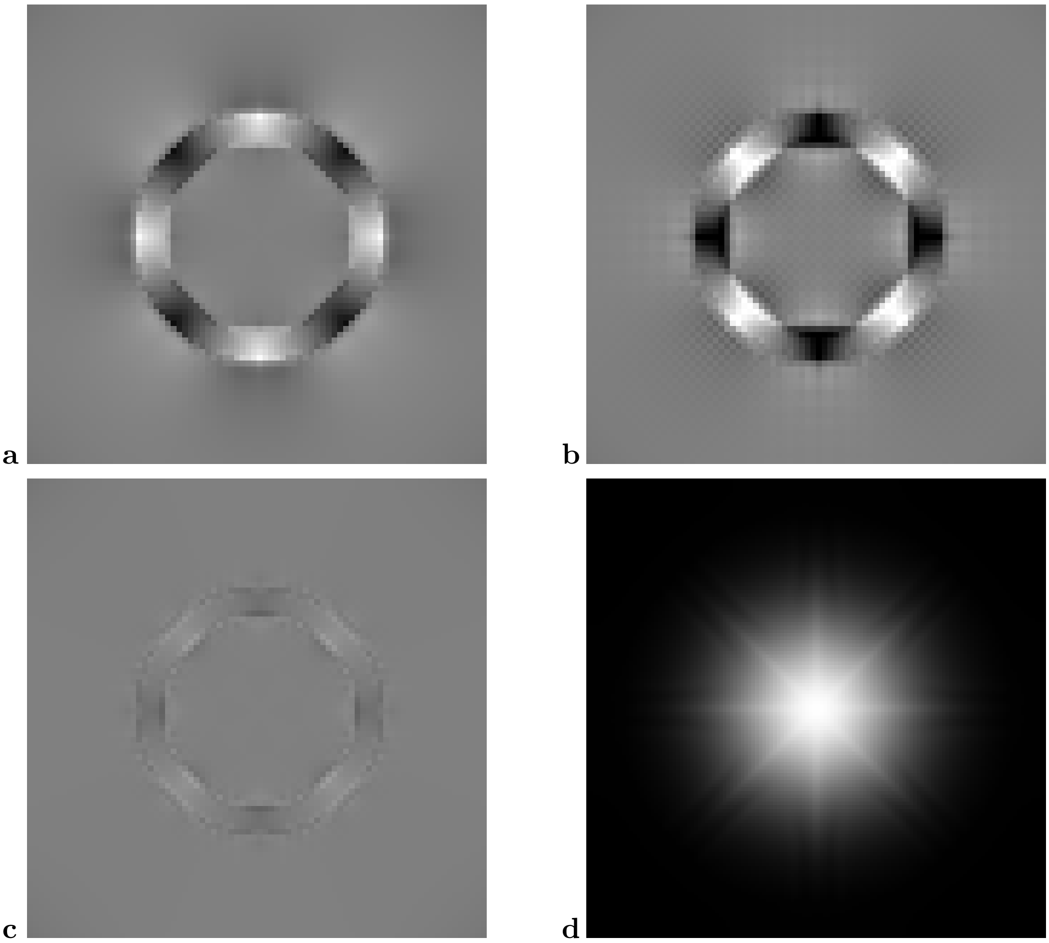


Figure 3.7: The artefacts on rotational symmetry.
(**a**–**c**) the (amplified) difference between a diffusion result from Figure 3.6, $\tau = 1$, and its copy rotated by 45 degrees: (**a**) splitting 1, (**b**) splitting 2, (**c**) splitting 3.
(**d**) Star-like patterns appear on a Gaussian diffused with a small diffusivity parameter λ (for any splitting method 1–3). Here $T = 200$, $\tau = 10$, the Perona-Malik λ was computed as the 10^{th} percentile of image gradients.

elements of the diffusion tensor $\boldsymbol{D} = \left(\begin{smallmatrix} a & b \\ b & c \end{smallmatrix}\right)$, is nonempty. Moreover, we have derived the formulas for the directional diffusivities depending on a single diffusivity parameter, and demonstrated experimentally that the directional splitting reveals better properties (regarding e.g. rotational symmetry and sensitivity to the time step size) if the splitting parameter is chosen from the interior of the admissible interval.

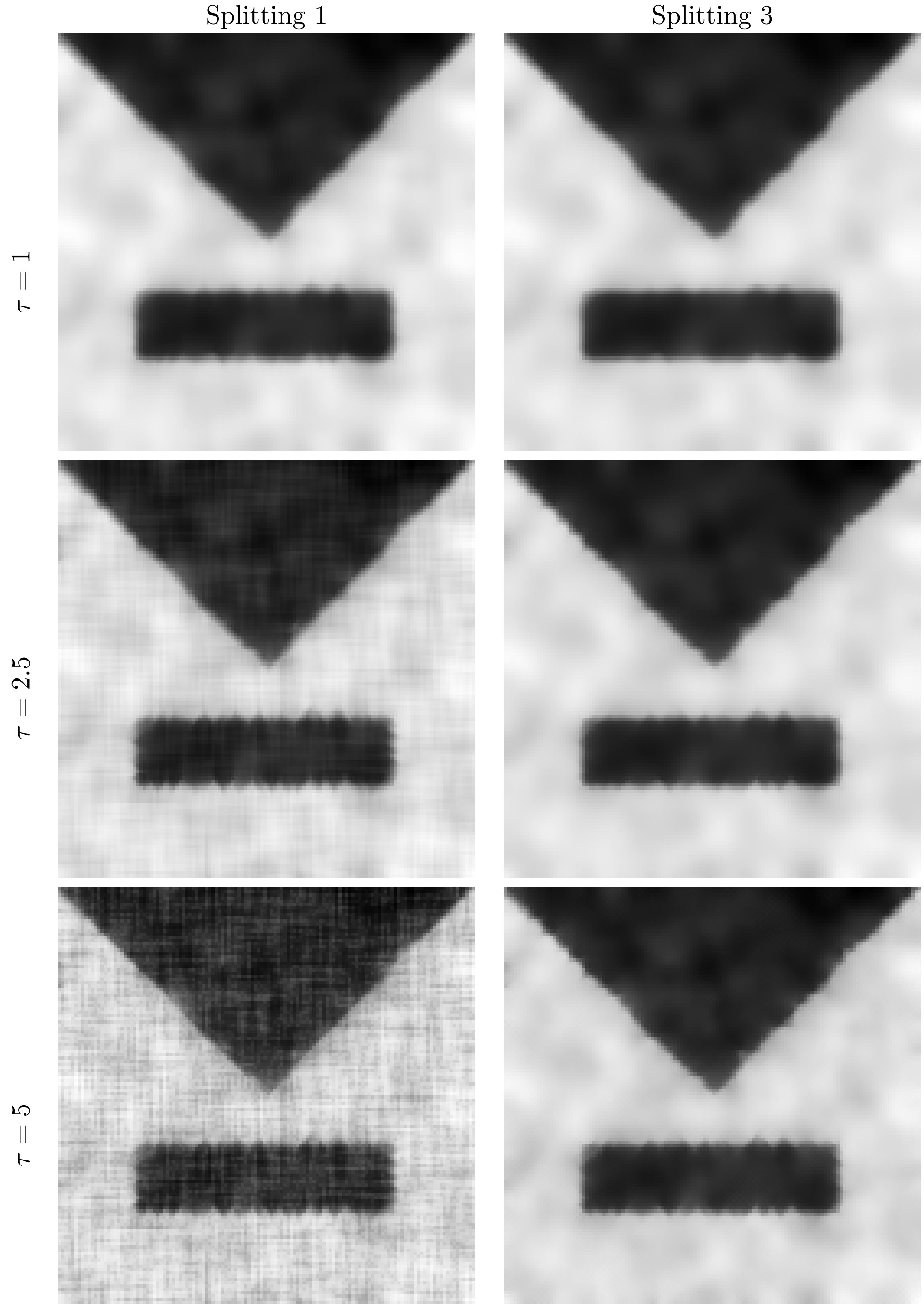


Figure 3.8: Impact of the directional splitting on the results of the noise-filtering procedure: splitting 1 on the left, splitting 3 on the right. In all cases, the stopping time of the diffusion was $T = 25$, the time step τ increases from top to bottom, $\tau \in \{1, 2.5, 5\}$.

3.3 Diffusion filtering algorithm

In this section, we summarize the ideas on numerical methods and parameter selection mentioned above into a complete and autonomous image filtering algorithm.

Let again $\widetilde{\mathbf{f}}$ stand for the ideal, noise-free data; our input image $\mathbf{f}$ is consists of the data and some additive noise n, $\mathbf{f} = \widetilde{\mathbf{f}} + n$. We assume that the noise has zero mean value, $E(n) = 0$, and that it is uncorrelated with the signal $\widetilde{\mathbf{f}}$. We want to find a filtered image $\mathbf{u}(T)$ which contains as much information of the ideal $\widetilde{\mathbf{f}}$ as possible. To achieve this goal, we propose to filter the data by the following algorithm of anisotropic nonlinear diffusion.

Algorithm 1: Adaptive diffusion image filtering

1. Initialize the image data, $\mathbf{u}(0) = \mathbf{f}$.

2. Initialize the diffusion parameters

a. user specified parameters

		possible values	*suggested value*
σ	noise scale (regularization, presmoothing)	$(0, \infty)$	1
φ_2	anisotropy	$[0, 1]$	0.2
ϱ	integration scale	$[0, \infty)$	0
τ_0	initial value for the iteration time step	$(0, \infty)$	1
p	parameter for the Perona–Malik estimation of λ	$(0, 1)$	$(0.7, 0.95)$

b. autonomously determined

λ the diffusivity parameter

If the noise n is known to be of normal distribution, estimate λ using the Black–Sapiro formula (2.47); otherwise employ the Perona–Malik procedure and set λ equal to the $(100 \cdot p)$-th percentile of regularized image gradients.

τ the iteration time step

Using the initial value τ_0 for τ, compute the first two iterations of the diffusion process (see below), $\mathbf{u}(\tau)$ and $\mathbf{u}(2\tau)$.
If $\operatorname{corr}\big(\mathbf{u}(0) - \mathbf{u}(\tau), \mathbf{u}(\tau)\big) \le \operatorname{corr}\big(\mathbf{u}(0) - \mathbf{u}(2\tau), \mathbf{u}(2\tau)\big)$ (see eq. (3.3)), the signal–noise correlation is not decreasing, the value of τ is too large. Set e.g. $\tau = \tau/4$, and repeat the two diffusion steps from $\mathbf{u}(0)$ with the current value of τ until the correlation decreases. Initialize $T = 2\tau$.

3. Iterate the diffusion filtering: repeat the diffusion filtering step from $\mathbf{u}(T)$ to $\mathbf{u}(T + \tau)$ until the minimum of the signal–noise correlation,

$$\operatorname{corr}\big(\mathbf{u}(0) - \mathbf{u}(T), \mathbf{u}(T)\big) < \operatorname{corr}\big(\mathbf{u}(0) - \mathbf{u}(T + \tau), \mathbf{u}(T + \tau)\big).$$

When we reach the minimum, the filtered image $\mathbf{u}(T)$ is found.

One iteration of the anisotropic diffusion process consists of the following steps (assuming the current data in $\mathbf{u}$, and storing the diffusion result to $\mathbf{o}$; see sections 2.4 and 2.6 for details on the procedure).

Algorithm 2: Anisotropic diffusion iteration

1. Calculate the structure tensor $\boldsymbol{S}_\varrho$

- **a.** Obtain the regularized image $\mathbf{s} = \mathbf{u}_\sigma$.
- **b.** Compute the gradient of the smoothed image, $\begin{pmatrix} d_x \\ d_y \end{pmatrix} = \nabla \mathbf{s}$ (using finite differences).
- **c.** Form the zero-order structure tensor, $\boldsymbol{S}_0 = \begin{pmatrix} {d_x}^2 & d_x \cdot d_y \\ d_x \cdot d_y & {d_y}^2 \end{pmatrix}$.
- **d.** If the integration scale ϱ is nonzero, calculate a component-wise convolution $\boldsymbol{S}_\varrho = G_\varrho * \boldsymbol{S}_0$.

2. Transform $\boldsymbol{S}_\varrho$ into the diffusion tensor $\boldsymbol{D}$

- **a.** Get eigenvectors $\vec{v}_1, \vec{v}_2$ and eigenvalues μ_1, μ_2 of the matrix $\boldsymbol{S}_\varrho$; assemble the eigenvectors columnwise into a matrix $\boldsymbol{X}$.
- **b.** If needed, this is the point where to estimate the diffusivity parameter λ from the array of eigenvalues μ_1.
- **c.** Calculate the diffusion tensor $\boldsymbol{D} = \boldsymbol{X} \begin{pmatrix} g(\mu_1) & 0 \\ 0 & \varphi_2 \end{pmatrix} \boldsymbol{X}^T$ where φ_2 is the parameter of anisotropy.

3. AOS diffusion scheme

- **a.** Separate the diffusion tensor into four diffusion directions, find the directional diffusivities α_l, $l = -1, \ldots, 2$ (using splitting 3 with equations (3.16)–(3.17)); initialize the output, $\mathbf{o}_{i,j} = 0$, $\forall i, j$.
- **b.** For all directions l, for each row in that direction
  - extract a row of pixels, $\mathbf{u}_l$, from the image $\mathbf{u}$
  - form the diffusivity matrix $\boldsymbol{A}_l$ from the diffusivities α_l corresponding to the row $\mathbf{u}_l$
  - construct one row of the output for the direction l by solving the equation $\mathbf{o}_l = (\boldsymbol{I} - 4\tau \boldsymbol{A}_l)^{-1} \mathbf{u}_l$. The matrix to be inverted is tridiagonal and the result can be computed efficiently using Thomas algorithm [53].
- **c.** Assemble the results of all directions, $\mathbf{o} = \frac{1}{4} \sum_{l=-1}^{2} \mathbf{o}_l$.

---

4 Monotonicity-enhancing NL diffusion

4.1 Introduction

Consider the following situation: let $\widetilde{\mathbf{f}}$ be a piecewise continuous real function defined on a rectangle $\Omega = [0, x_{\max}] \times [0, y_{\max}] \subset \mathbb{R}^2$. Moreover, let $\widetilde{\mathbf{f}}$ be *piecewise monotone*, i.e. the domain Ω can be partitioned into K connected subsets $\Omega_k, k = 1, \ldots, K$, such that $\widetilde{\mathbf{f}}$ is continuous and monotone on Ω_k:

$$\begin{aligned} &\widetilde{\mathbf{f}}(x_1, y) \leq \widetilde{\mathbf{f}}(x_2, y) && \forall (x_1, y), (x_2, y) \in \Omega_k, \;\; x_1 < x_2 \\ \text{or } &\widetilde{\mathbf{f}}(x_1, y) \geq \widetilde{\mathbf{f}}(x_2, y) && \forall (x_1, y), (x_2, y) \in \Omega_k, \;\; x_1 < x_2 \end{aligned} \tag{4.1}$$

$$\begin{aligned} &\widetilde{\mathbf{f}}(x, y_1) \leq \widetilde{\mathbf{f}}(x, y_2) && \forall (x, y_1), (x, y_2) \in \Omega_k, \;\; y_1 < y_2 \\ \text{or } &\widetilde{\mathbf{f}}(x, y_1) \geq \widetilde{\mathbf{f}}(x, y_2) && \forall (x, y_1), (x, y_2) \in \Omega_k, \;\; y_1 < y_2 \end{aligned} \tag{4.2}$$

For discrete data $\mathbf{f}$, obtained from $\widetilde{\mathbf{f}}$ using the discretization formula (2.1), the piecewise monotonicity assumption can be restated as follows: if K is the smallest number of connected[1] sets Ω_k needed to partition the function domain so that the discrete function $\mathbf{f}$ is continuous[2] and monotone on each Ω_k, we require that K is much smaller than the number of pixels in the image. The discretization noise may violate this monotonicity assumption; the gradient of the noisy samples $f_{i,j}$ will change its orientation much more often than that of the original function. Our task is to restore the desired function properties, filter the noise, smooth or simplify the sampled function $\mathbf{f}$ so as to enforce the piecewise monotonicity (reduce the number K described above) while preserving important discontinuities or edges.

The piecewise monotonicity is intended as a weakening or generalization of piecewise linearity. Let us present three real-world examples which motivate and

[1] Using 4–neighbourhood to define discrete connectedness.

[2] Obviously, the notion of function continuity does not exist in the discrete situation. The gradient (more exactly its estimate from the discrete data) may serve as a replacement: the smaller the gradient at a given position, the more feasible it is to regard the function as continuous around that position.

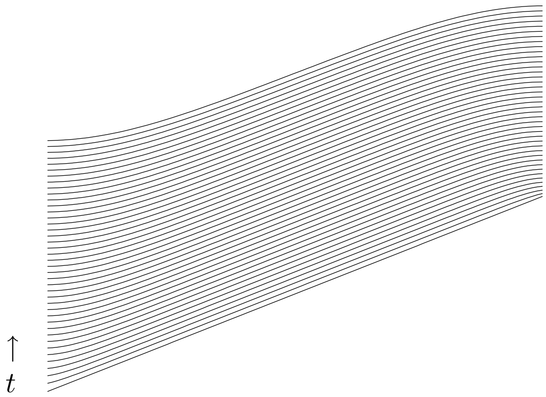


Figure 4.1: Classical nonlinear diffusion approaches a (piecewise) constant function; this phenomenon can be observed first near the ends of growing function segments.

clarify this notion. First, the rainbow range finder which was already introduced on page 2: the coordinate axes of the data measured by this device are fixed by the illumination conditions, and the wavelength of light observed in the scene is continuous on continuous object surfaces, and piecewise monotone, more precisely piecewise increasing in one coordinate direction. It is possible to formulate a similar property for other types of range finders (using laser rays, structured light, etc.) as well.

As a second example of a situation where the piecewise monotonicity assumption could be appropriate, we may mention the range data for 3D reconstruction in computer vision. The function $\widetilde{\mathbf{f}}$ represents the distance of the objects in the scene from the camera; the distance changes gradually on continuous surfaces, with abrupt changes, i.e. discontinuities where a different object comes into view. These distance data are measured at discrete positions x_i, y_j with some imprecision modeled by the noise n, thus forming a 2D array (or image) of values $f_{i,j}$.

As a third example, some black-and-white images can be considered piecewise monotone if the grey values or light intensities change gradually in some parts of the image, with possible discontinuities (i.e. edges) between image regions.

In this chapter we concentrate on the possibility to enhance piecewise monotonicity by nonlinear diffusion. Unfortunately, as the classical nonlinear diffusion filters presented in the previous chapters smooth the data more inside homogeneous regions, the function $\mathbf{u}$ solving the diffusion equation tends to a piecewise constant as time t increases. This is illustrated on a simple function in Figure 4.1; the 'horizontalization' of an increasing function can be observed first near the ends of continuous function segments. While nonlinear diffusion yields impressive results on some images and may be particularly useful for robust image segmentation, the model assuming piecewise constancy is unsuitable for noise removal from most natural scenes.

A variety of other possible models for image filtering, such as piecewise linear, locally monotone or locally convex has been suggested in [1, 2]. The limitations

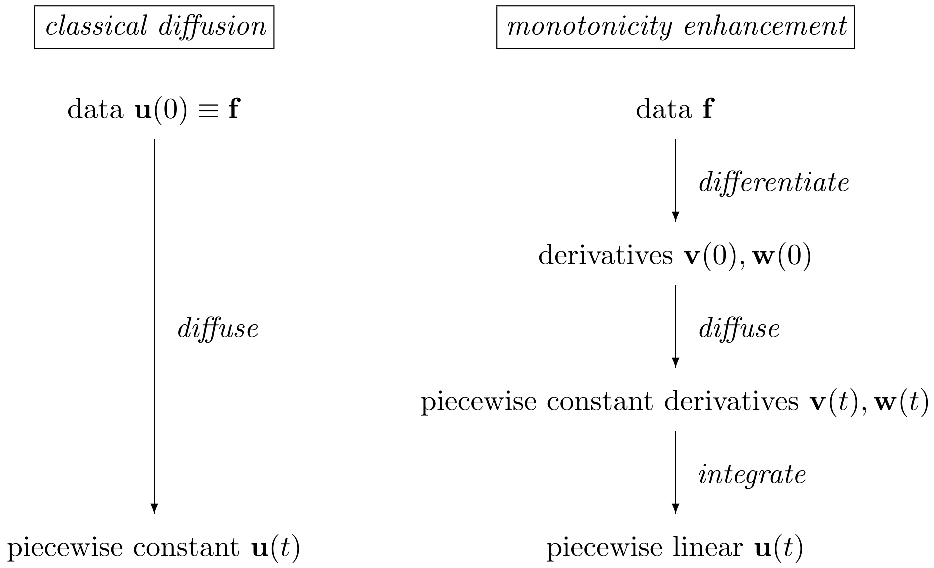

classical diffusion
monotonicity enhancement
data $\mathbf{u}(0) \equiv \mathbf{f}$
data $\mathbf{f}$
differentiate
derivatives $\mathbf{v}(0), \mathbf{w}(0)$
diffuse
diffuse
piecewise constant derivatives $\mathbf{v}(t), \mathbf{w}(t)$
integrate
piecewise constant $\mathbf{u}(t)$
piecewise linear $\mathbf{u}(t)$

Figure 4.2: Main idea of the monotonicity enhancement by nonlinear diffusion: running the NL diffusion on directional derivatives of the original data and integrating the result, we obtain a filtered output which tends to a piecewise linear function.

of the piecewise constant model for noise removal using minimization of the total variation of the image[3] were observed by Chambolle and Lions in [11]. The authors suggest to alleviate the problem by introducing second order terms (like the total variation of the image gradient) into the functional to be minimized. A coupled diffusion of partial derivatives, partly similar to the approach forming the main topic of this chapter, has been proposed by Whitaker and Gerig in [55]; their main motivation was to use the boundaries of the diffused derivatives (of some order N) to make decisions about presence of higher order image features.

In this chapter we try to exploit the following simple idea: if we differentiate the data first and run the diffusion on the arrays of partial derivatives instead of the original function values, the piecewise smoothing of derivatives leads (after integration) to data which are piecewise monotone, for higher t approaching a piecewise linear function. Where classical nonlinear diffusion simplifies an image into segments of similar grey levels, the procedure using derivatives (the main topic of this chapter; we will call it *monotonicity-enhancing nonlinear diffusion*, abbreviated ME NL diffusion) creates patches of similar *trends*, which can successfully approximate a large class of images and other types of inputs and will be preferred if piecewise monotonicity of the data is assumed and desired.

In the following, we first present an isotropic nonlinear diffusion algorithm for monotonicity enhancement as developed in [20, 21]. The method approximates directional derivatives by one-sided differences of neighbouring pixels. This approach keeps the image information contained in the derivatives located close to

[3] We should remark that image restoration by minimization of some functional is closely related to nonlinear diffusion and scale-space theory, see e.g. [28, 45, 31, 34] and Section 2.3.3.

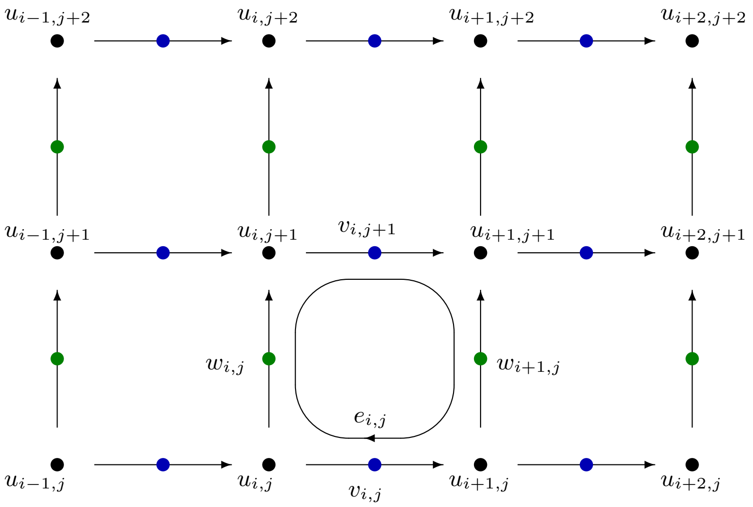


Figure 4.3: Geometry of the discrete image: the partial derivatives of the data $\mathbf{u}$ in the direction of axes x, y are approximated by finite differences forming arrays $\mathbf{v}$ and $\mathbf{w}$, respectively.

the original positions, but leads to complications as the elements of the arrays of the derivatives and of the array of diffusivity information do not overlap. For more complex anisotropic nonlinear diffusion filters, central differences for derivative approximation become advantageous. This latter method is described in Section 4.3. We then discuss theoretical properties of the methods, and present experimental results.

4.2 Isotropic ME NL diffusion

4.2.1 From data to partial derivatives

Consider the two-dimensional situation: a function $\mathbf{f}(x, y)$ is sampled and represented by values $f_{i,j}$ at positions (x_i, y_j), $x_i = i \cdot \Delta x$, $y_j = j \cdot \Delta y$, $i = 1, \ldots, N_i$, $j = 1, \ldots, N_j$; the samples (pixels) $f_{i,j}$ form an image, our input data.

The partial derivatives of the original function in the direction of axes x, y, respectively, can be approximated from the discrete image by differences of the neighbouring pixels, forming two arrays $\mathbf{v}$ and $\mathbf{w}$:

$$\left.\frac{\partial \mathbf{f}(x,y)}{\partial x}\right|_{x_i,y_j} \approx v_{i,j} \equiv \frac{f_{i+1,j} - f_{i,j}}{\Delta x}, \qquad i = 1, \ldots, N_i - 1,\ j = 1, \ldots, N_j, \qquad (4.3)$$

$$\left.\frac{\partial \mathbf{f}(x,y)}{\partial y}\right|_{x_i,y_j} \approx w_{i,j} \equiv \frac{f_{i,j+1} - f_{i,j}}{\Delta y}, \qquad i = 1, \ldots, N_i,\ j = 1, \ldots, N_j - 1. \qquad (4.4)$$

The arrays $\mathbf{v}, \mathbf{w}$ must satisfy some requirements in order to represent the partial derivatives of a real function. In the continuous domain, the integral of a function gradient along any closed curve is zero,

$$\oint_{\mathcal{C}} \nabla \mathbf{f}(\vec{r}) \cdot \mathrm{d}\vec{r} = 0. \tag{4.5}$$

In the discrete image any closed curve is composed of the elementary closed curves passing through four pixels[4] as illustrated in Figure 4.3, and the observation (4.5) transforms into

$$\forall i,j: \quad e_{i,j} \equiv w_{i,j} + v_{i,j+1} - w_{i+1,j} - v_{i,j} = 0. \tag{4.6}$$

While this constraint is satisfied automatically by (4.3)–(4.4), we have to be more careful about it during the diffusion process.

4.2.2 The diffusion algorithm

Rewriting the isotropic nonlinear diffusion equation (2.71) for the arrays $\mathbf{v}, \mathbf{w}$ instead of $\mathbf{u}$, we obtain the following set of equations:

$$\partial_t \mathbf{v} = \operatorname{div}\left(g_v(\mathbf{v}, \mathbf{w})\, \nabla \mathbf{v}\right) \tag{4.7}$$

$$\partial_t \mathbf{w} = \operatorname{div}\left(g_w(\mathbf{v}, \mathbf{w})\, \nabla \mathbf{w}\right) \tag{4.8}$$

These two equations could be solved separately from each other; however, the information in the two fields $\mathbf{v}$ and $\mathbf{w}$ is not entirely independent, and we choose to couple the two equations by a common array of diffusivities as usual for vector-valued diffusion (see [55, 47, 16]). In the continuous settings, the diffusivities g_v and g_w (computed using both the images $\mathbf{v}, \mathbf{w}$ as the notation $g(\mathbf{v}, \mathbf{w})$ suggests) will be identical.

The discrete algorithms for these equations can be developed analogically to a single isotropic diffusion equation; we refer the reader to Section 2.6 for details. As an example, the explicit discretization scheme will have the following form:

$$\mathbf{v}^{k+1} = \Big(I + \tau \boldsymbol{A}_x(\mathbf{v}^k, \mathbf{w}^k) + \tau \boldsymbol{A}_y(\mathbf{v}^k, \mathbf{w}^k)\Big)\, \mathbf{v}^k \tag{4.9}$$

$$\mathbf{w}^{k+1} = \Big(I + \tau \boldsymbol{B}_x(\mathbf{v}^k, \mathbf{w}^k) + \tau \boldsymbol{B}_y(\mathbf{v}^k, \mathbf{w}^k)\Big)\, \mathbf{w}^k \tag{4.10}$$

Here $\mathbf{v}, \mathbf{w}$ are column vectors of generally different size, $(N_i - 1) \cdot N_j$, and $N_i \cdot (N_j - 1)$, respectively; the matrices $\boldsymbol{A}$, $\boldsymbol{B}$ which store the diffusivity information have also different dimensions. However, as the connection between elements $v_{i-1,j}$ and $v_{i,j}$, and between $w_{i,j-1}$ and $w_{i,j}$ passes through a common point ($u_{i,j}$ in Figure 4.3), we find it reasonable to assign these two connections the same diffusivity, namely $g_{i,j}$. This way the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ depend on both $\mathbf{v}^k$ and $\mathbf{w}^k$; the equations (4.9), (4.10) will be coupled through the common array of diffusivities and many elements of $\boldsymbol{A}_x$ and $\boldsymbol{B}_y$ will be identical (similarly for the other pair of directions and $\boldsymbol{A}_y$, $\boldsymbol{B}_x$, only the common point of the connections $v_{i,j}$ — $v_{i,j+1}$ and $w_{i,j}$ —

[4] Assuming 4–neighbourhood, i.e. only vertical and horizontal connections are allowed.

$w_{i+1,j}$ does not coincide with any input datum $\mathbf{u}$; we denote the diffusivity of that position by $g_{i+\frac{1}{2},j+\frac{1}{2}}$).

There are many possibilities how to assemble the information from the smoothed, regularized versions $\tilde{\mathbf{v}}, \tilde{\mathbf{w}}$ of the arrays $\mathbf{v}, \mathbf{w}$ into the common diffusivities $g_{i,j}$. Proceeding most directly from (2.71) to the diffusion of derivatives, we obtain the following:

$$g\left(|\nabla^2 \mathbf{u}_\sigma|^2\right) = g\left(|\nabla \cdot (\nabla \mathbf{u}_\sigma)|^2\right) \approx \; g\left(|\nabla \cdot (\tilde{\mathbf{v}}, \tilde{\mathbf{w}})|^2\right) = g\left(\left|\frac{\tilde{\mathbf{v}}}{\partial x} + \frac{\tilde{\mathbf{w}}}{\partial y}\right|^2\right) \approx$$
$$g_{i,j} \equiv g\left(|\tilde{v}_{i,j} - \tilde{v}_{i-1,j} + \tilde{w}_{i,j} - \tilde{w}_{i,j-1}|^2\right). \tag{4.11}$$

In our experiments, we used the equation (2.46) to define the function g.

The complication with this approach is that derivatives amplify high frequency components of a signal (including noise), and the second order derivatives of the input data which appear in formula (4.11) make the method more difficult to tune and unsuitable for highly corrupted inputs. In some of our experiments we employed the following trick successfully: steer the diffusion not with a gradient of the partial derivatives, but with a gradient of the original data, thus avoiding higher order derivatives. Using the arrays of partial derivatives we can write

$$g\left(|\nabla \mathbf{u}_\sigma|^2\right) \approx \qquad g_{i,j} \equiv g\left(\left|\frac{\tilde{v}_{i-1,j} + \tilde{v}_{i,j}}{2}\right|^2 + \left|\frac{\tilde{w}_{i,j-1} + \tilde{w}_{i,j}}{2}\right|^2\right), \tag{4.12}$$
$$g_{i+\frac{1}{2},j+\frac{1}{2}} \equiv g\left(\left|\frac{\tilde{v}_{i,j} + \tilde{v}_{i,j+1}}{2}\right|^2 + \left|\frac{\tilde{w}_{i,j} + \tilde{w}_{i+1,j}}{2}\right|^2\right).$$

This alternative reveals a drawback, too: limited to first derivatives only, it may neglect discontinuities of the second derivatives and round corners of a continuous function. See some experiments below.

There is another problem with the simple formulation of nonlinear diffusion of partial derivatives: the equations (4.7)–(4.8) do not guarantee that the constraint (4.6) is satisfied. In the remaining part of this section we try to enforce the necessary properties of the arrays of partial derivatives.

Denote by $\hat{\mathbf{z}} = [\mathbf{v}^k, \mathbf{w}^k]^T$ the result of the diffusion process at time t_k. We seek a solution $\mathbf{z}$ as close as possible to $\hat{\mathbf{z}}$ while obeying the constraint (4.6) which can be written in matrix form as

$$\boldsymbol{C}\,\mathbf{z} = 0 \tag{4.13}$$

where $\boldsymbol{C}$ is a $[(N_i - 1) \cdot (N_j - 1)] \times N$ sparse matrix with four nonzero entries in each row, and N is the total number of elements of the arrays $\mathbf{v}, \mathbf{w}$:

$$N = (N_i - 1) \cdot N_j + N_i \cdot (N_j - 1). \tag{4.14}$$

The rigorous way to solve this problem is to find $\mathbf{z}$ as an orthogonal projection of $\hat{\mathbf{z}}$ into the null space S of matrix $\boldsymbol{C}$; such solution minimizes the norm $\|\mathbf{z} - \hat{\mathbf{z}}\|_2$. The following details are adapted from Luenberger [18].

If we find the vectors forming the orthonormal basis of the null space S of $\boldsymbol{C}$ and arrange them as columns of a matrix $\boldsymbol{D}$, then satisfying $\boldsymbol{C}\,\mathbf{z} = 0$ corresponds to evaluating $\mathbf{z} = \boldsymbol{D}\delta$ for some vector δ, which we find as the one minimizing

$$\|\boldsymbol{D}\,\delta - \hat{\mathbf{z}}\|_2. \tag{4.15}$$

As the columns of $\boldsymbol{D}$ are orthonormal, the equation (4.15) is minimized for $\delta = \boldsymbol{D}^T\,\hat{\mathbf{z}}$, and the sought orthogonal projection of $\hat{\mathbf{z}}$ into the null space of matrix $\boldsymbol{C}$ is

$$\mathbf{z} = \boldsymbol{D}\boldsymbol{D}^T\,\hat{\mathbf{z}}. \tag{4.16}$$

The advantage is that $\boldsymbol{D}\boldsymbol{D}^T$ can be computed only once for the same image size, the matrix $\boldsymbol{C}$ remains the same for all diffusion steps.

The trouble with this mathematically correct solution it that it involves the construction of an orthonormal basis of the null space of $\boldsymbol{C}$, and full matrix multiplication (processes of complexities $O(N^3)$ and $O(N^2)$, respectively). Already for small images, these matrix computations become infeasible.

As a viable alternative, we choose to restore the property (4.6) by the following iterative algorithm:

Algorithm 3: Restoration of the derivatives

1. Evaluate errors

$$e_{i,j} = w_{i,j} + v_{i,j+1} - w_{i+1,j} - v_{i,j}, \quad \forall i,j.$$

2. For all i,j, update the values as follows[5]:

$$v_{i,j} = v_{i,j} + (e_{i,j} - e_{i,j-1})/c, \qquad w_{i,j} = w_{i,j} - (e_{i,j} - e_{i-1,j})/c$$

with obvious modifications at the image boundary.

3. If $\max |e_{i,j}|$ is smaller than a given threshold θ, finish; otherwise go to 1.

Similarly to the vector $\hat{\mathbf{z}}$ (originating from $\mathbf{v},\mathbf{w}$), also the errors $\hat{e}_{i,j}$ can be organized into one vector, $\hat{\mathbf{e}}$. It may be computed by the formula $\hat{\mathbf{e}} = \boldsymbol{C}\hat{\mathbf{z}}^k$, where $\hat{\mathbf{z}}^k$ is the current (k-th iteration) solution of Algorithm 3.

The space $S^\perp$, orthogonal to the null space S, is exactly the linear hull of all rows of the matrix $\mathbf{C}$. The orthogonal projection of $\hat{\mathbf{z}}$ onto S is the unique vector $\mathbf{z} \in S$ such that $\mathbf{z} - \hat{\mathbf{z}} \in S^\perp$. Algorithm 3 starts from $\hat{\mathbf{z}}^0 \equiv \hat{\mathbf{z}}$; in each step of the algorithm, we add to $\hat{\mathbf{z}}^k$ a linear combination of rows of $\boldsymbol{C}$, i.e., a vector from $S^\perp$. Thus we remain in the hyperplane orthogonal to S and containing $\mathbf{z}$. The linear combination is nonzero iff $\hat{\mathbf{e}}$ has a nonzero entry. This implies that the only fixed point of the algorithm is $\mathbf{z}$. As a consequence, whenever Algorithm 3 converges, its limit is $\mathbf{z}$, the orthogonal projection of $\hat{\mathbf{z}}$ into the null space of matrix $\boldsymbol{C}$.

[5] For simplicity, we use the same symbols for the original and the updated values, using a MATLAB-like notation; new values are on the left.

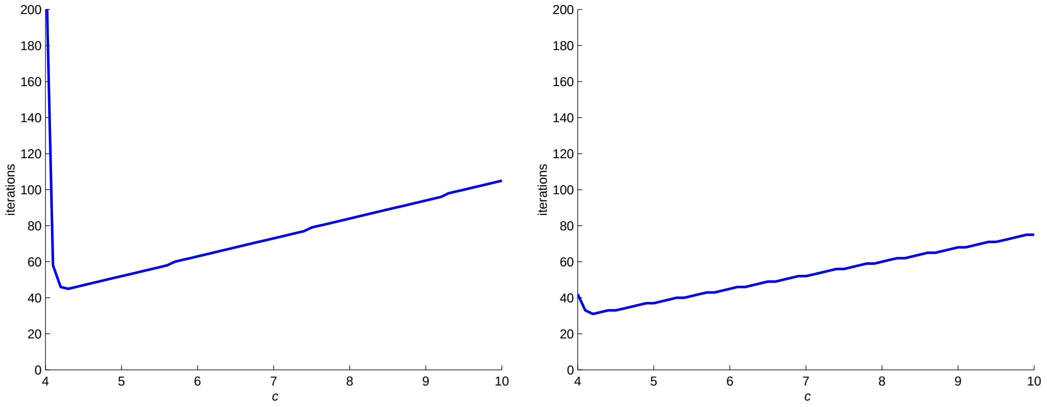


Figure 4.4: Number of iterations needed to complete the derivative-correction algorithm for precision $\theta = 0.001$ on a 20×20 grid, plotted against the division constant c. Left: the input was a basis function $\hat{\mathbf{z}}^l$ with nonzero entry in the center of array $\mathbf{v}$; right: nonzero entry at a corner of $\mathbf{v}$. The algorithm converges for any $c \geq 4$, the fastest convergence is obtained with $c \approx 4.3$.

The choice of the constant c determines the length of the vector added to $\hat{\mathbf{z}}^k$ in each step. It influences the convergence and its speed. We can analyse the convergence of Algorithm 3 and find the optimal value for c using linearity.

Let $\hat{\mathbf{z}} = [\mathbf{v}^k, \mathbf{w}^k]^T$ stand again for the result of the diffusion process at time t_k. Any $\hat{\mathbf{z}}$ can be written as a linear combination of basis functions $\hat{\mathbf{z}}^l = [\hat{z}_i^l]$, $i = 1, \ldots, N$, with zeros at all but one position:

$$\hat{z}_i^l = \begin{cases} 1 & i = l, \\ 0 & \text{otherwise.} \end{cases} \tag{4.17}$$

Then, if $\widetilde{\mathbf{z}}$ is the result of the derivative-correcting algorithm with $\hat{\mathbf{z}}$ as the input, it can be composed as a linear combination (with the same coefficients as for $\hat{\mathbf{z}}$) of functions $\widetilde{\mathbf{z}}^l$ where $\widetilde{\mathbf{z}}^l$ is the result of the algorithm correcting the data $\hat{\mathbf{z}}^l$. Therefore, to verify convergence, it is sufficient to consider the basis functions $\hat{\mathbf{z}}^l$.

To be able to compare the iterative restoration results with the correct least square solution, we performed an experiment on a small 20×20 grid, for which we were able to evaluate equation (4.16). We obtained the following results:

- The derivative-restoration algorithm converges for any $c \geq 4$.
  The constant c in the algorithm divides the errors into the elements which contributed to it. The value $c = 4$ is a reasonable choice as four elements form each $e_{i,j}$; a slightly higher number ($c \approx 4.3$) damps down oscillations and leads to a faster convergence (see Fig. 4.4).

- The iteratively restored $\widetilde{\mathbf{z}}$ converges to the solution of equation (4.16) (i.e. to the orthogonal projection of the diffused derivatives into the null space of matrix $\boldsymbol{C}$) as the threshold θ decreases to zero.

The complexity of one iteration of this algorithm is only $O(N)$. Although several iterations are necessary, this still compares favorably with the complexity of the matrix method.

4.2.3 From derivatives back to data

Let us now assume that the filtered arrays $\mathbf{v}, \mathbf{w}$ are available, and that they contain correct values in the sense of condition (4.6). These arrays contain (redundantly) all the information needed for the reconstruction of the image $\mathbf{u}$ up to a scalar u_0 added to function values. The integration can be performed by the following algorithm:

```
ALGORITHM 4: INTEGRATION OF u FROM v, w

1. Reconstruct the first row:
      $\hat{u}_{1,1} = 0$
      for $i = 2, \ldots, N_i$ : $\hat{u}_{i,1} = \hat{u}_{i-1,1} + v_{i-1,1}$
2. Reconstruct the columns:
      for $i = 1, \ldots, N_i$
         for $j = 2, \ldots, N_j$: $\hat{u}_{i,j} = \hat{u}_{i,j-1} + w_{i,j-1}$
3. Fix the shift of function values:
      for all $i, j$ : $u_{i,j} = \hat{u}_{i,j} + u_0$
```

The choice of the scalar u_0 influences significantly the behaviour of the filtering procedure as a whole. One possibility is to select it so that the average grey values of the original, $\mathbf{f}$, and the filtered image, $\mathbf{u}$, remain equal:

$$u_0 = \frac{1}{N_i \cdot N_j} \left[\sum_{i=1}^{N_i} \sum_{j=1}^{N_j} f_{i,j} - \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} \hat{u}_{i,j} \right] . \tag{4.18}$$

Alternatively, on the condition that the range of the filtered data is not larger than that of the original function,

$$\max \hat{\mathbf{u}} - \min \hat{\mathbf{u}} \leq \max \mathbf{f} - \min \mathbf{f}, \tag{4.19}$$

it is possible to select u_0 so that the monotonicity-enhancing procedure satisfies the maximum-minimum principle (2.30). If that is the desired behaviour, we require

$$u_0 \in [\min \mathbf{f} - \min \hat{\mathbf{u}},\ \max \mathbf{f} - \max \hat{\mathbf{u}}]. \tag{4.20}$$

If the average-value u_0 from equation (4.18) falls into these bounds, the procedure can fulfill both the grey-value invariance and the maximum-minimum principle; however, these two conditions are contradictory for some data (see Theorem 4.2 on page 62).

4.3 Anisotropic ME NL diffusion

As well as the classical isotropic diffusion, the isotropic diffusion of derivatives reveals the drawback of leaving the noise unfiltered near the locations where the diffusivity is inhibited. These locations include the edges and discontinuities of the image function, and newly also the corners, i.e. positions of higher curvature, or higher magnitude of the second derivatives of the data. In analogy to the classical case, the performance of the monotonicity-enhancing diffusion at filtering of such noise can be improved by moving from isotropic to anisotropic diffusion.

The diffusion of derivatives is extended into the anisotropic filter similarly as the classical diffusion. The only trouble is that we will again need the diffusivity computed for several diffusion directions from the diffusion tensor; the computations become again more complex in comparison to the isotropic filter, and our representation from the previous section, using arrays of different size for each of the directional derivatives $\mathbf{v}, \mathbf{w}$ (and consequently differently sized diffusivity matrices for the discrete computations) becomes rather unsuitable for the new situation. Instead, we will use centralized finite differences to approximate the directional derivatives,

$$\frac{\partial \mathbf{f}(x,y)}{\partial x}\Big|_{x_i,y_j} \approx v_{i,j} \equiv \frac{f_{i+1,j} - f_{i-1,j}}{2\,\Delta x}, \quad i = 2,\dots,N_i - 1,\; j = 1,\dots,N_j, \tag{4.21}$$

$$\frac{\partial \mathbf{f}(x,y)}{\partial y}\Big|_{x_i,y_j} \approx w_{i,j} \equiv \frac{f_{i,j+1} - f_{i,j-1}}{2\,\Delta y}, \quad i = 1,\dots,N_i,\; j = 2,\dots,N_j - 1 \tag{4.22}$$

(combined with one-sided differences at the image border), and make the discrete arrays $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$ all have the same dimensions $N_i \times N_j$.

This arrangement results in a new geometry of the discrete image (see Figure 4.5) with the advantage that the arrays of derivatives now share the positions of their elements with each other and with the original array of the data $\mathbf{u}$. Also, the arrays of diffusivities (understood either as the diffusion tensor $\boldsymbol{D}$ or separated into the directional diffusivities α_k) have the same dimensions $N_i \times N_j$, and their elements are found at the same locations as the elements of $\mathbf{u}$. The only disadvantage of using the central differences lies in the fact that more distant data elements are now directly connected by the derivatives, and the filtering tends to blur the image a little bit more than the isotropic filter using one-sided differences.

4.3.1 Algorithms

In this section we summarize the updates and changes involved in the shift from one-sided to central differences to approximate the directional derivatives, and those linked to the extension of the isotropic filter to the anisotropic one.

The first step of the procedure, the conversion from the data $\mathbf{u}$ to the directional derivatives $\mathbf{v}, \mathbf{w}$, is defined by equations (4.21)–(4.22) above (except for the boundary pixels for which one-sided differences are employed).

The second step, the vector-valued anisotropic nonlinear diffusion of the direc-

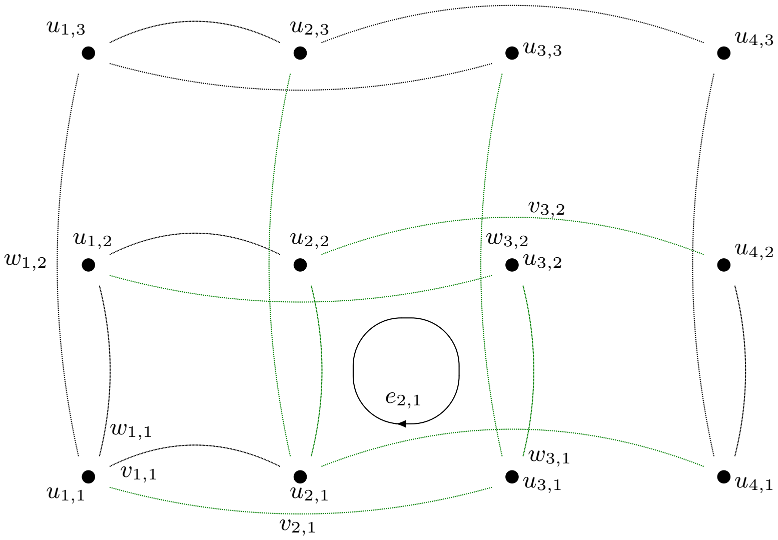


Figure 4.5: Geometry of the lower left corner of a discrete image for the anisotropic monotonicity-enhancing diffusion. The partial derivatives of the data $\mathbf{u}$ in the direction of axes x, y are approximated by finite differences forming arrays $\mathbf{v}$ and $\mathbf{w}$, respectively; the elements $u_{i,j}$, $v_{i,j}$ and $w_{i,j}$ all share the same position (black circle in the picture). As an example, the lines corresponding to the derivatives which contribute to the error cycle $e_{2,1}$ are colored in green (see the anisotropic error redistribution algorithm below.)

tional derivatives, can be expressed using the following set of equations:

$$\partial_t \mathbf{v} = \operatorname{div}\left(\boldsymbol{D}(\mathbf{v}, \mathbf{w})\, \nabla \mathbf{v}\right) \tag{4.23}$$

$$\partial_t \mathbf{w} = \operatorname{div}\left(\boldsymbol{D}(\mathbf{v}, \mathbf{w})\, \nabla \mathbf{w}\right). \tag{4.24}$$

Analogically to the isotropic case, these two equations will be coupled through a common diffusion tensor $\boldsymbol{D}(\mathbf{v}, \mathbf{w})$, but unlike the situation of the one-sided differences in Section 4.2.2, their discrete counterparts will share the same structure of the data. These anisotropic diffusion equations will be solved in the same way as the classical anisotropic diffusion (see sections 2.4 and 2.6.5); the only difference lies in the way the structure tensor $\boldsymbol{S}_0$ is constructed from the data, newly integrating the gradient information from both channels $\mathbf{v}, \mathbf{w}$:

$$\boldsymbol{S}_0 = \frac{1}{2}\left(\nabla \mathbf{v}_\sigma\, \nabla \mathbf{v}_\sigma^T + \nabla \mathbf{w}_\sigma\, \nabla \mathbf{w}_\sigma^T\right). \tag{4.25}$$

The rest of the procedure (forming $\boldsymbol{D}$ from $\boldsymbol{S}$, separating it into one-directional diffusivities, etc.) remains identical to the classical anisotropic diffusion.

A final remark on the diffusion part: having two arrays of derivatives $\mathbf{v}$ and $\mathbf{w}$ we performed a vector-valued diffusion where the individual components of the data were unified by the common field of diffusivity information. Note that this situation is very easy to extend to colour or other images consisting of several information channels, so we can readily obtain algorithms which are able either to filter colour data by the classical (piecewise constant) diffusion, either isotropic or anisotropic, and to develop the method further into a monotonicity enhancing diffusion of colour images. We will present some experiments later on.

The third step of the monotonicity-enhancing procedure consists in enforcing the gradient field constraint (4.5). The reasoning remains unchanged from Section 4.2.2; for central differences, the iterative correction algorithm assumes the following form.

Algorithm 5: Restoration of the derivatives (for central differences)

1. Evaluate errors

$$e_{i,j} = \big(w_{i,j} + v_{i,j+1} - w_{i+1,j} - v_{i,j} + w_{i,j+1} + v_{i+1,j+1} - w_{i+1,j+1} - v_{i+1,j}\big)/2$$

$\forall i = 1, \ldots, N_i - 1,\ j = 1, \ldots, N_j - 1$. (See the error cycle in Figure 4.5).

2. For all i, j, update the values as follows:

$$\begin{aligned} v_{i,j} &= v_{i,j} + (e_{i-1,j+1} - e_{i-1,j-1} + e_{i,j+1} - e_{i,j-1})/c, \\ w_{i,j} &= w_{i,j} - (e_{i+1,j-1} - e_{i-1,j-1} + e_{i+1,j} - e_{i-1,j})/c, \end{aligned}$$

$c \geq 4$, with obvious modifications at the image boundary.

3. If $\max |e_{i,j}|$ is smaller than a given threshold, finish; otherwise go to 1.

Note that leaving the derivatives unconstrained during the whole diffusion process and restoring the derivatives afterwards gives more influence to the restoration procedure at the final stage of the filtering algorithm. We prefer to restore the derivatives after each step, so that the corrected derivatives can be iteratively updated by the model of the diffusion equations. It seems that this latter alternating solution slightly restricts the blurring of the image caused by the restoration algorithm which spreads the errors in all directions.

Lastly, we come to the integration of the data from the diffused directional derivatives:

Algorithm 6: Integration of $\mathbf{u}$ from central differences $\mathbf{v}, \mathbf{w}$

```
1  Reconstruct the first row:
       $\hat{u}_{1,1} = 0$
       for $i = 2, \ldots, N_i$ : $\hat{u}_{i,1} = \hat{u}_{i-1,1} + (v_{i-1,1} + v_{i,1})/2$
2. Reconstruct the columns:
       for $i = 1, \ldots, N_i$
           for $j = 2, \ldots, N_j$: $\hat{u}_{i,j} = \hat{u}_{i,j-1} + (w_{i,j-1} + w_{i,j})/2$
3. Fix the shift of function values:
       for all $i, j$ : $u_{i,j} = \hat{u}_{i,j} + u_0$.
```

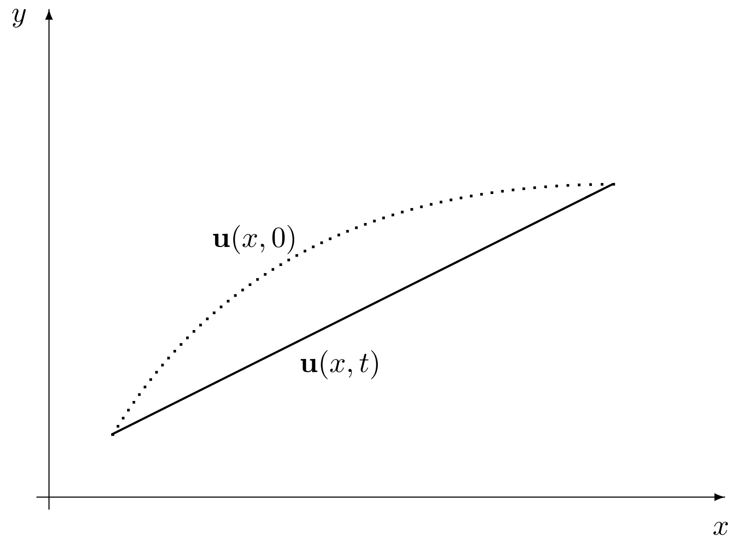


Figure 4.6: For some data, the maximum-minimum principle contradicts the conservation of average value.

4.4 Theoretical properties

Theorem 4.1. *Let* $\mathbf{u}(x,t)$ *be the result of the monotonicity-enhancing diffusion at time* t*. For* $t \to \infty$*,* $\mathbf{u}(x,t)$ *converges to a linear function.*

Proof. For each of the directional derivatives $\mathbf{v}, \mathbf{w}$, the diffused result $\mathbf{v}(x,t)$, $\mathbf{w}(x,t)$ approaches a constant function as t increases (see Section 2.3.4). By integration, the constant derivatives $\mathbf{v}(x,t)$, $\mathbf{w}(x,t)$ lead to a linear function $\mathbf{u}(x,t)$; this fact is not effected by the derivative-restoration algorithm. □

Theorem 4.2. *For some data* $\mathbf{u}$*, the monotonicity-enhancing diffusion cannot satisfy both the maximum-minimum principle and the conservation of average grey level.*

Proof. The maximum-minimum principle was defined in equation (2.30); using a modified notation, it reads

$$\inf_{x\in\Omega} \mathbf{u}(x,0) \leq \mathbf{u}(x,t) \leq \sup_{x\in\Omega} \mathbf{u}(x,0) \tag{4.26}$$

where the first parameter of $\mathbf{u}$ is the spatial position, and the second denotes the diffusion time. The average grey level is conserved if

$$\frac{1}{|\Omega|} \int_{\Omega} \mathbf{u}(x,t)\,\mathrm{d}x = \mu, \qquad \forall t. \tag{4.27}$$

We can now prove this theorem by having a look at a simple 1D example in Figure 4.6. If the arc was the original function $\mathbf{u}(x,0)$, the straight line can represent the solution to which the diffusion result $\mathbf{u}(x,t)$ converges for $t \to \infty$. Because NL diffusion conserves the average value of the diffused function, the diffused directional derivative in the monotonicity-enhancing diffusion must have the same average value as the derivative of the original data; after integration and for this particular (monotonic) example, the size of the interval of function values of the diffused $\mathbf{u}(x,t)$ is the same as that of the original $\mathbf{u}(x,0)$. More precisely, let x_1 and x_2 be the left and right extreme of the domain of $\mathbf{u}$; then $\mathbf{u}(x_1,0) - \mathbf{u}(x_2,0) = \mathbf{u}(x_1,t) - \mathbf{u}(x_2,t)$.

If we want to satisfy the extremum principle, the straight line has to lie within the y-interval defined by the extreme points of the original arc; however, in this position the whole line remains below the arc, and the average values of the two object differ. If, on the other hand, we tried to fix the two objects to the same average value, the line would extend above the arc at one end, thus violating the maximum-minimum principle. Obviously, this reasoning applies not only to the solution at infinity (the straight line), but to any intermediate result $\mathbf{u}(x,t)$. □

Although the maximum-minimum principle and the conservation of average grey-level may be incompatible, we did not experience this conflict in experiments with piecewise continuous noisy data as presented in Section 5.2. Moreover, the experiments suggest that the range of the filtered data spans a smaller interval than the range of the noisy function, so – omitting the grey-level conservation – it should be always possible to choose the integration constant u_0 such that the maximum-minimum principle holds.

If the choice between the maximum-minimum principle and the average grey-level has to be made, the decision will depend on the application in mind. The absolute level of image intensities might play some role e.g. in medical applications where the grey-values correspond to some physical measurements; in image processing, the variety of possible changes of illumination, camera sensitivity or aperture, etc., will often make the average grey-level unreliable, and preference may be given to the maximum-minimum principle.

Corollary 4.3. *Let again $\mathbf{u}(x,t)$ be the result of the monotonicity-enhancing diffusion at time t. For $t \to \infty$, the number K of monotone regions of $\mathbf{u}(x,t)$ converges to 1.*

Proof. We introduced the number K in Section 4.1 as the smallest number of sets needed to partition the function into monotone pieces. It follows directly from Theorem 4.1 that this number converges to one as $t \to \infty$. □

Remark: this descent of K is monotone for one-dimensional data. In 2D, new separated monotone regions may appear (similarly to new local extrema which are created even by the linear, Gaussian diffusion process) and the number K does not need to decrease monotonically towards convergence. An example in Figure 4.7 shows the number K for several first iterations of the monotonicity-enhancing anisotropic diffusion for the noisy 2D data of Figure 5.12. Although difficult to see, there are parts in this plot where K increases (around $t = 15$ and $t = 35$).

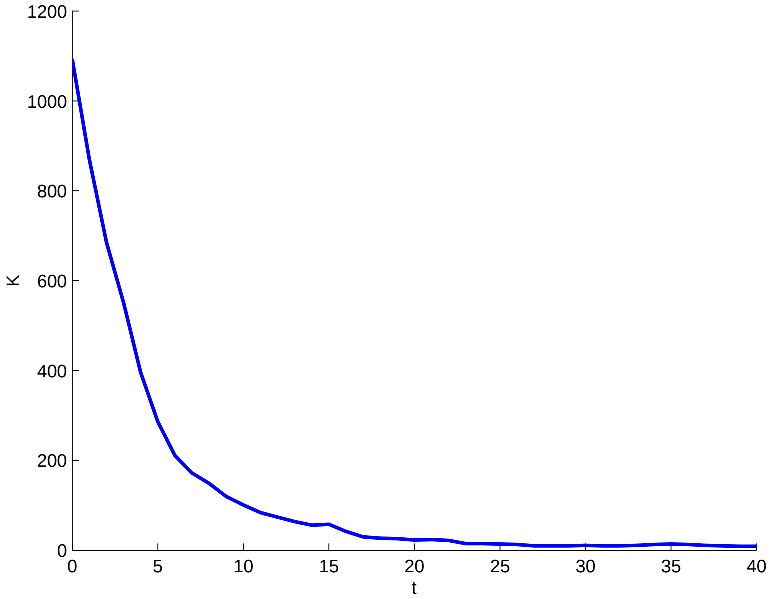


Figure 4.7: The number K developing with the diffusion time. K is the number of sets needed to partition the function domain into regions such that the data are monotone on each such region. This particular example employed anisotropic monotonicity-enhancing diffusion to filter the data of Figure 5.12b.

5 Experiments

5.1 Image filtering with anisotropic NL diffusion

This section presents the experiments we performed to test the ability of the anisotropic nonlinear diffusion to remove additive noise from image data. The filtering method is based on the anisotropic NL diffusion of Section 2.4 and the additive operator splitting (AOS) discretization scheme; the actually employed parts of the methods and the ideas on parameter selection from Section 2.3.5 are summarized into a complete and autonomous image filtering algorithm in Section 3.3.

The results we obtained confirm that anisotropic nonlinear diffusion is a powerful image restoration method (the best we know of), particularly suitable for piecewise constant data and additive noise[1].

5.1.1 Triangle and rectangle experiment

This section presents an experiment comparing the results of different diffusion algorithms filtering an originally black and white image with non-Gaussian additive noise. We also demonstrate the influence of some parameters on the anisotropic diffusion filtering; the effect of various numerical splitting procedures was shown on the same data in Section 3.2.

The input data are shown in Figure 5.1: the noisy image was obtained by adding noise of uniform distribution in the range $[-255, 255]$ to the ideal input, and by restricting the noisy values into the interval $[0, 255]$.

In Figure 5.2, the noise is smoothed by linear diffusion, isotropic nonlinear diffusion, and two anisotropic diffusion filters of Section 2.6.5; the grey-values are stretched to the whole interval $[0, 255]$ so that a higher contrast between the dark

[1] The anisotropic diffusion has been extended to restore shift-invariantly and shift-variantly blurred images with noise by You and Kaveh in [57]. The degradation process is also described by equation (2.1), but then their procedure tries both to remove the noise n and reconstruct the convolution kernel h. Our method is based on the assumption that the discretization by h acts locally, its blurring effect can be omitted and only the additive noise n has to be filtered.

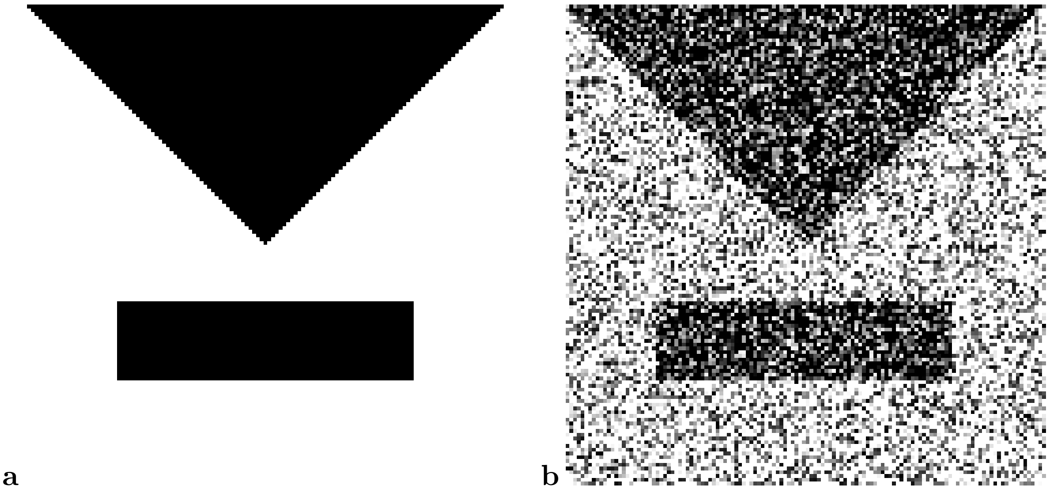


Figure 5.1: Ideal and noisy input image ($[0, 127]^2 \rightarrow [0, 255]$) for the 'Triangle and rectangle' experiment. Noise with uniform distribution in the range $[-255, 255]$ was added to the two-valued synthetic data.

and bright regions corresponds to a better noise-filtering performance. The stopping time was determined autonomously by the *decorrelation criterion* described by the equation (3.4); see Section 3.1. You can see that in all cases, although quite different filtering algorithms were employed, the stopping criterion leads to results where most of the noise is removed and the ideal signal becomes apparent or suitable for further processing; we support this statement by showing the thresholded content of the filtered images in Figure 5.3.

The stopping criterion was designed to minimize the MAD distance from the ideal function. If visual quality was the goal to be achieved, we would probably stop the diffusion later, especially as linear diffusion (Figure 5.2a) and the anisotropic diffusion with maximum $\varphi_2 = 1$ (Figure 5.2c) are concerned. We find however that the MAD distance and the visual quality are in a good agreement in Figure 5.2d which represents exactly the result of the diffusion filtering algorithms as proposed in Section 3.3.

The effect of varying the anisotropic parameter[2] φ_2 is demonstrated in Figure 5.4. You can see that too small an anisotropy leaves noise at edges unfiltered (and the image resembles that obtained by the isotropic NL diffusion), while setting the level of φ_2 too high results in rounding of corners and blurring of image features. This latter drawback is caused by the bounds enforced on the condition number of the diffusion tensor $\boldsymbol{D}$: a high diffusivity in the coherence direction φ_2 requires a nonzero diffusivity φ_1 even if the diffusion in the direction of the maximum grey level change would be – without this bound – inhibited by the gradient

[2] More precisely, φ_2 is the second eigenvalue of the diffusion tensor (see Section 2.4.2) and controls the amount of diffusion in the direction of maximum coherence. At egdes where diffusion in the gradient direction is inhibited, a higher φ_2 means that the method is 'more anisotropic' (i.e. with different amounts of diffusion in different directions), and that is why we call φ_2 the parameter of anisotropy.

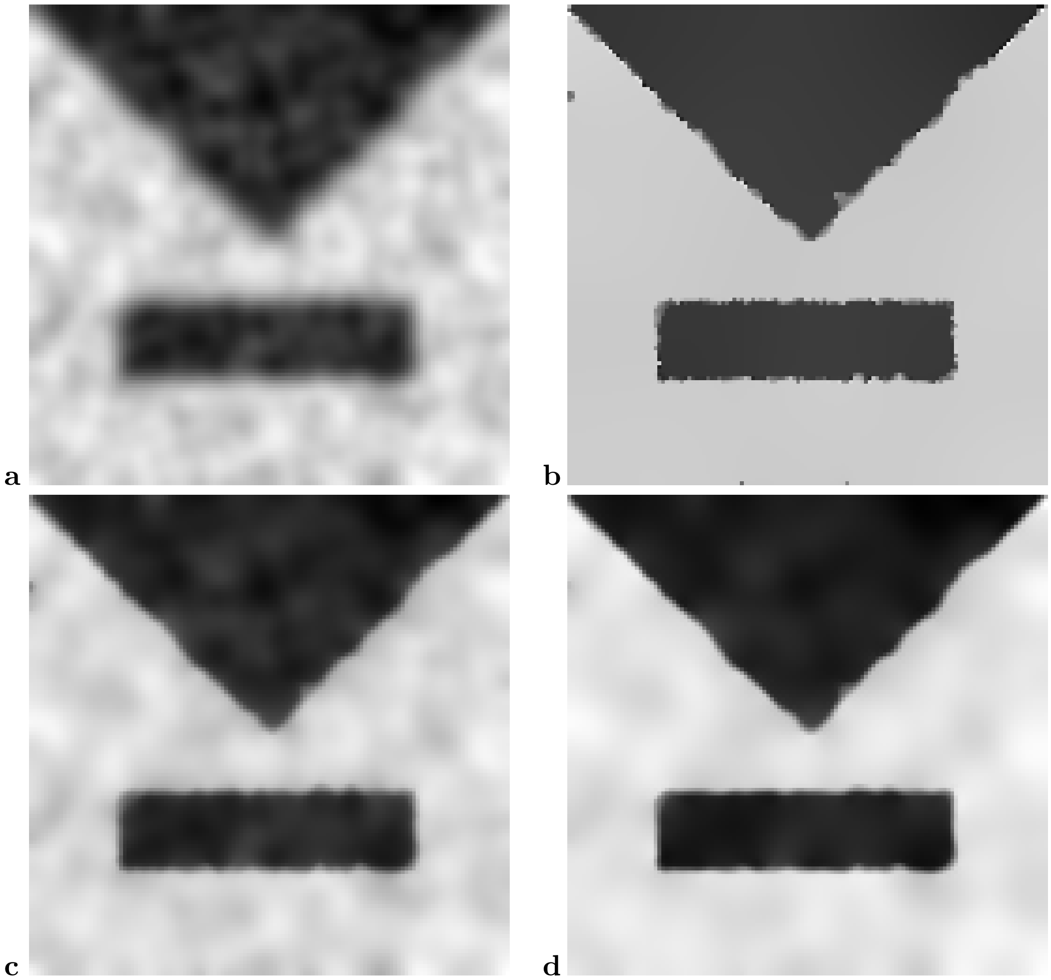


Figure 5.2: Comparing the different diffusion algorithms on the noisy data of Figure 5.1, all with the stopping time selected autonomously by minimizing the criterion (3.3): (a) linear diffusion, $T = 3.8$; (b) isotropic nonlinear diffusion, $T = 125$; (c) anisotropic NL diffusion, splitting 3, $\varphi_2 = 1$ (maximum anisotropy), $T = 15$; (d) anisotropic NL diffusion, splitting 3, $\varphi_2 = 0.2$, $T = 32$.
In (b)–(d), the parameters $\sigma = 1$, $\tau = 1$ were employed, and the parameter λ was estimated using the Perona-Malik procedure from $p = 0.9$ (i.e. as the 90^{th} percentile of regularized image gradients) in each step.

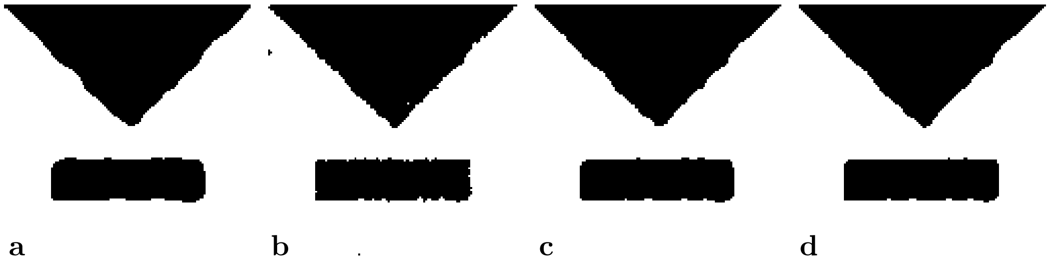


Figure 5.3: Thresholded versions of the images in Figure 5.2

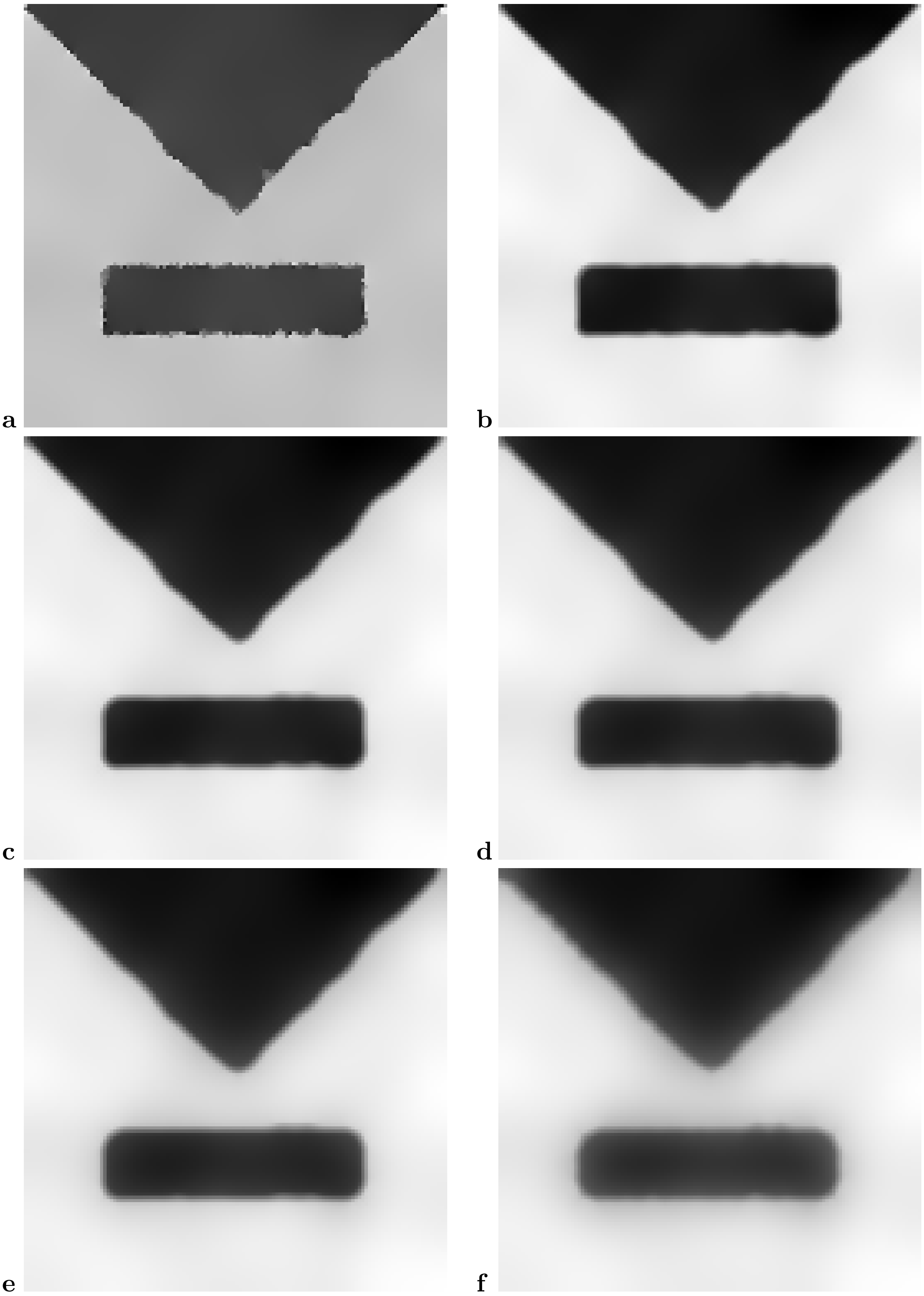


Figure 5.4: Effect of varying the anisotropy parameter φ_2 on the filtered result: (a)–(f) $\varphi_2 = 0,\ 0.1,\ 0.2,\ 0.3,\ 0.5,\ 1$, respectively. The remaining parameters were $T = 100$, $\tau = 2$, $\sigma = 1$, and λ was estimated using the Perona–Malik procedure from $p = 0.9$.

magnitude. To reduce both the corner-rounding and the edge-blurring side effects of anisotropic diffusion, we suggest to use smaller values for the parameter φ_2; any choice from $[0.1, 0.25]$ is still sufficient to filter the noise near edges while reducing the undesirable artefacts considerably.

5.1.2 Cymbidium experiment

Noise of normal distribution was added to the image of a cymbidium flower (Figure 5.5 top, courtesy of Michal Haindl), the noisy image was subjected to diffusion filtering algorithm as described in Section 3.3, with the parameters $\sigma = 1$, $\tau_0 = 0.5$, $\varphi_2 = 0.2$. The optimal values for the remaining parameters λ and T were estimated autonomously, using the Black–Sapiro's equation (2.47) for λ, and our decorrelation criterion (3.4) to determine the optimal T. The values of the parameters employed are shown together with the measured filtering results in Table 5.1. The filtering performance is evaluated by the MAD distance from the ideal data, so a smaller number means a better result. You can see that in all cases, the nonlinear diffusion filter combined with our time-selection strategy leads to results largely superior to median filter, and slightly better than the recursive filter [13][3]. Several noisy and filtered images can be seen in Figures 5.5 and 5.6.

Input image			MAD after filtering			diffusion params.	
Filename	SNR	MAD	median	recursive	diffusion	T	λ
p66n.pgm	48.95	0.45	2.07	0.03	0.76	1.0	8.58
p27n.pgm	31.02	2.41	2.60	2.3	1.76	2.5	8.49
p24n.pgm	28.10	3.21	2.85	3.1	2.05	3.0	8.55
p17n.pgm	19.95	7.32	4.37	5.4	3.20	6.0	26.15
p15n.pgm	17.96	8.97	5.04	5.8	3.55	7.0	26.51
p13n.pgm	14.87	12.33	6.43	6.6	4.24	8.5	35.97
p9n.pgm	9.99	20.84	10.08	8.2	5.93	10.5	77.56
p7n.pgm	8.41	25.10	11.91	8.8	6.79	11.0	119.37
p5n.pgm	6.17	33.62	15.59	10.5	8.38	11.5	192.79
p4n.pgm	4.69	42.14	19.26	13.0	9.99	12.0	284.34

Table 5.1: The cymbidium experiment. The first three columns give the filename of a given noisy image, the signal-to-noise ratio, and the MAD distance between the original data and the noisy input. The next three columns compare the filtering results of three methods: the 3x3 median filter, the Haindl's recursive filter [13], and the anisotropic NL diffusion. The last two columns present the actually employed diffusion parameters.

[3] Except the almost-no-noise case p66n.pgm; this exception was caused by different input data and discretization noise. The results of the recursive filter were kindly provided by Michal Haindl from the Academy of Sciences of the Czech Republic.

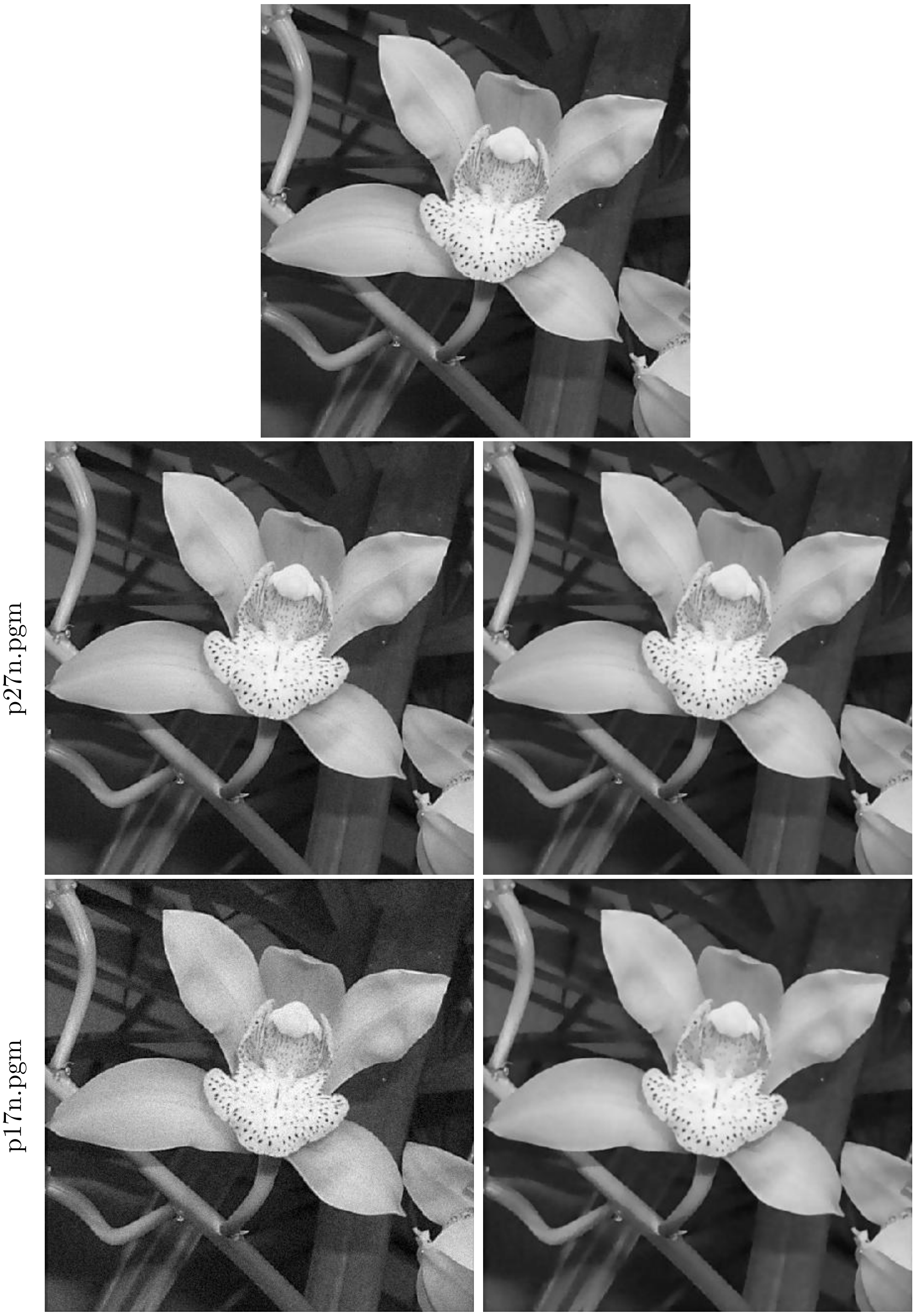


Figure 5.5: The cymbidium experiment. Top: original, noise-free image. Left column (top to bottom): input images p27n.pgm and p17n.pgm. Right column: corresponding images filtered by the anisotropic NL diffusion. The stopping time T was chosen autonomously using the decorrelation criterion; see Table 5.1 for parameter values and a quantitative evaluation of the filtering performance.

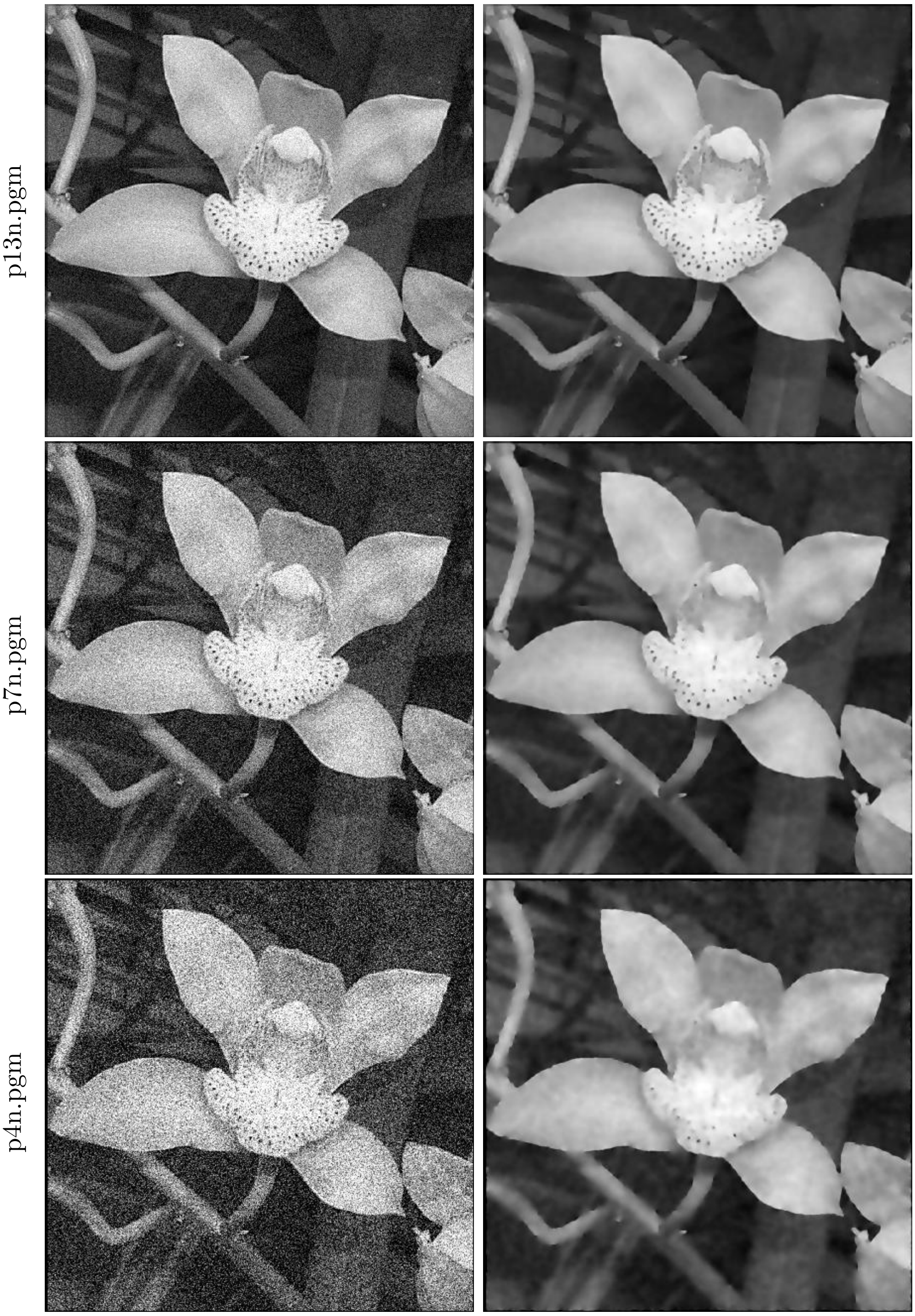


Figure 5.6: The cymbidium experiment continued. Left column (top to bottom): input images p13n.pgm, p7n.pgm, and p4n.pgm. Right column: corresponding images filtered by the anisotropic NL diffusion. See text and Table 5.1 for details.

5.1.3 Colour images

As mentioned in Section 2.5, the algorithms for NL diffusion can be easily extended to vector-valued data, including colour images. We present several filtering examples here.

First, Figure 5.7 shows the effect of the integration scale ϱ on the result of anisotropic diffusion. The results are obtained using the autonomous stopping criterion (3.4) and it also shows the limitations of that method: although the input image contains no visible corruptions, the high-frequency fur is erroneously considered as noise and oversmoothed in the filtered result. We might advocate the stopping time by saying that the input does not comply well with the piecewise constant assuption implicit in the diffusion equations; anyway, this is an example of a situation where it would be desirable to combine the decorrelation criterion with other information (e.g. the expected amount of noise in the input image) to decide on a better (i.e. earlier) stopping time. Another observation: the filtering result using the decorrelation stopping time (3.4) is in this case quite stable with respect to the amount of additive noise in the data. This fact is depicted in Figure 5.8; the main features become apparent in the filtered version and the smoothing may serve as a stabilizing factor for further processing.

So far, we tested the filtering capabilities of nonlinear diffusion mainly on additive noise. A different kind of image distortion is caused by jpeg compression. Especially when trying to keep the file size small, as is often needed e.g. for images to be presented on the internet, some oscilations appear in the jpeg-compressed data near edges. Also, if the image is subsampled, aliasing effects may appear. Nonlinear diffusion filtering is able to alleviate both these artefacts.

Let us have a look at the example at the top of Figure 5.9, taken from the www presentation of Škoda car manufacturer (`http://www.skoda-auto.cz/`). Typical jpeg oscilations are visible e.g. inside the windscreen, an aliasing example at the wheel rim or at the door. If we want to produce higher quality pictures, we can supersample the image and smooth it using anisotropic diffusion. One filtering result is seen at the bottom of Figure 5.9; both aliasing and jpeg oscilations are reduced substantially. Several details of the original and the filtered image are shown for comparison in Figure 5.10.

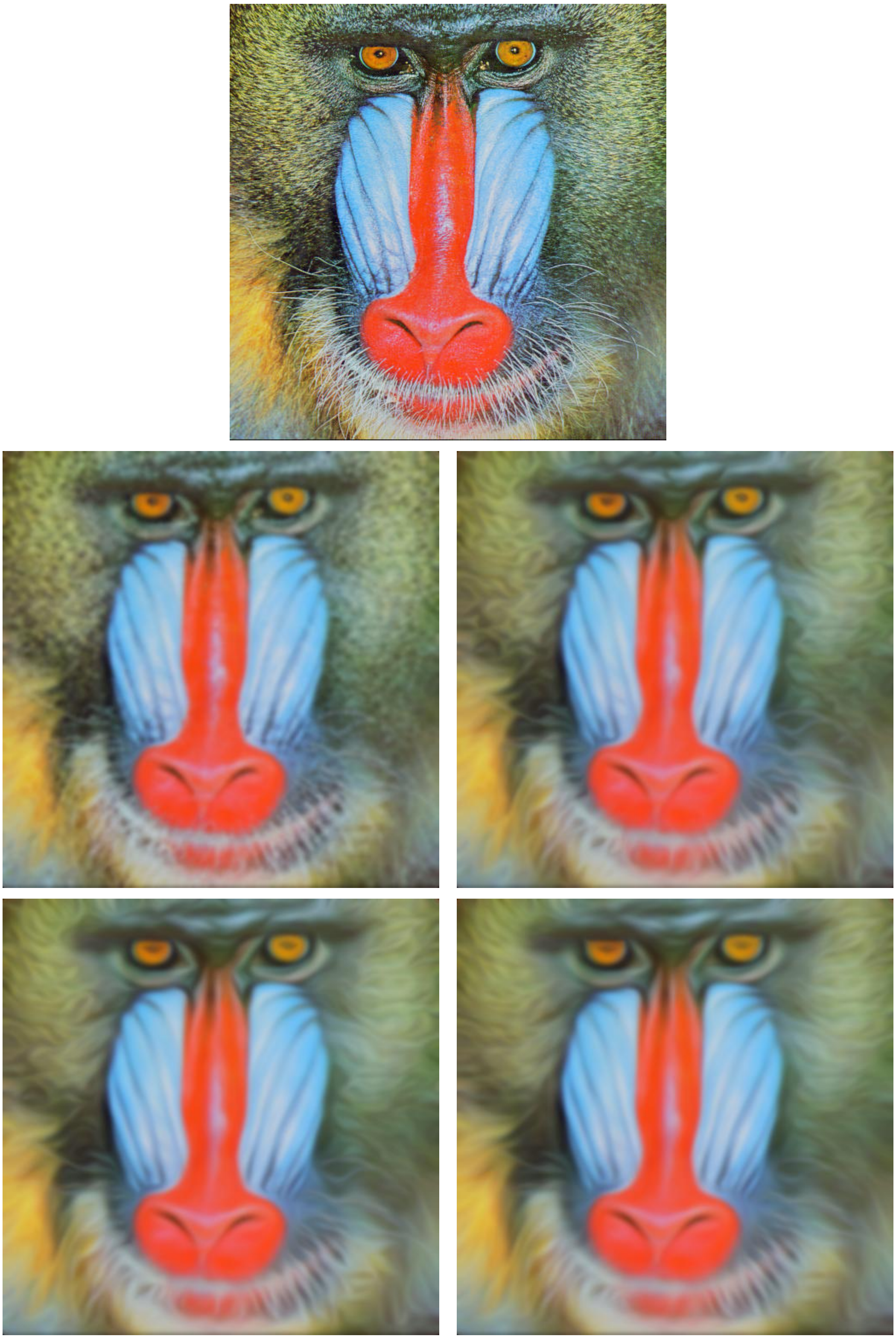


Figure 5.7: Top: input 512 × 512 image of a baboon. This image was filtered with anisotropic diffusion with the following parameters: $\sigma = 1$, $\varphi_1 = g(\mu_1)$ with $\lambda = 0.001$, $\varphi_2 = 1$. The integration parameter ϱ was varied (center to bottom and left to right): $\varrho = 0$, 3, 5, 10, and the criterion (3.4) lead to stopping times $T = 45$, 50, 51, 51, respectively.

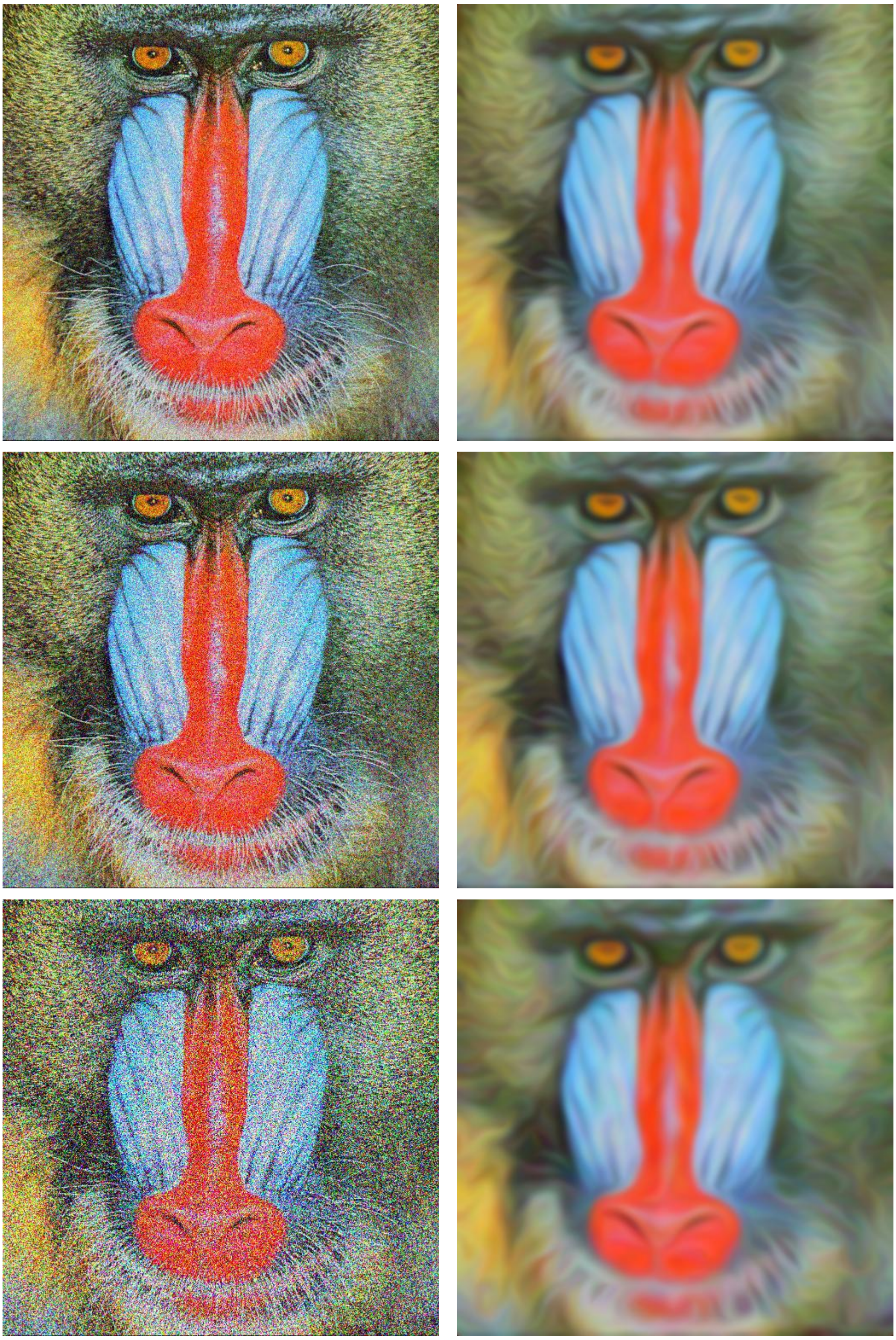


Figure 5.8: Left: baboon with additive Gaussian noise of variance (top to bottom) 30, 50, 100. Right: filtered with anisotropic diffusion using $\varrho = 3$ and autonomous decorrelation time $T = 52$, 61, 98, respectively. The remaining parameters were set as in Figure 5.7.

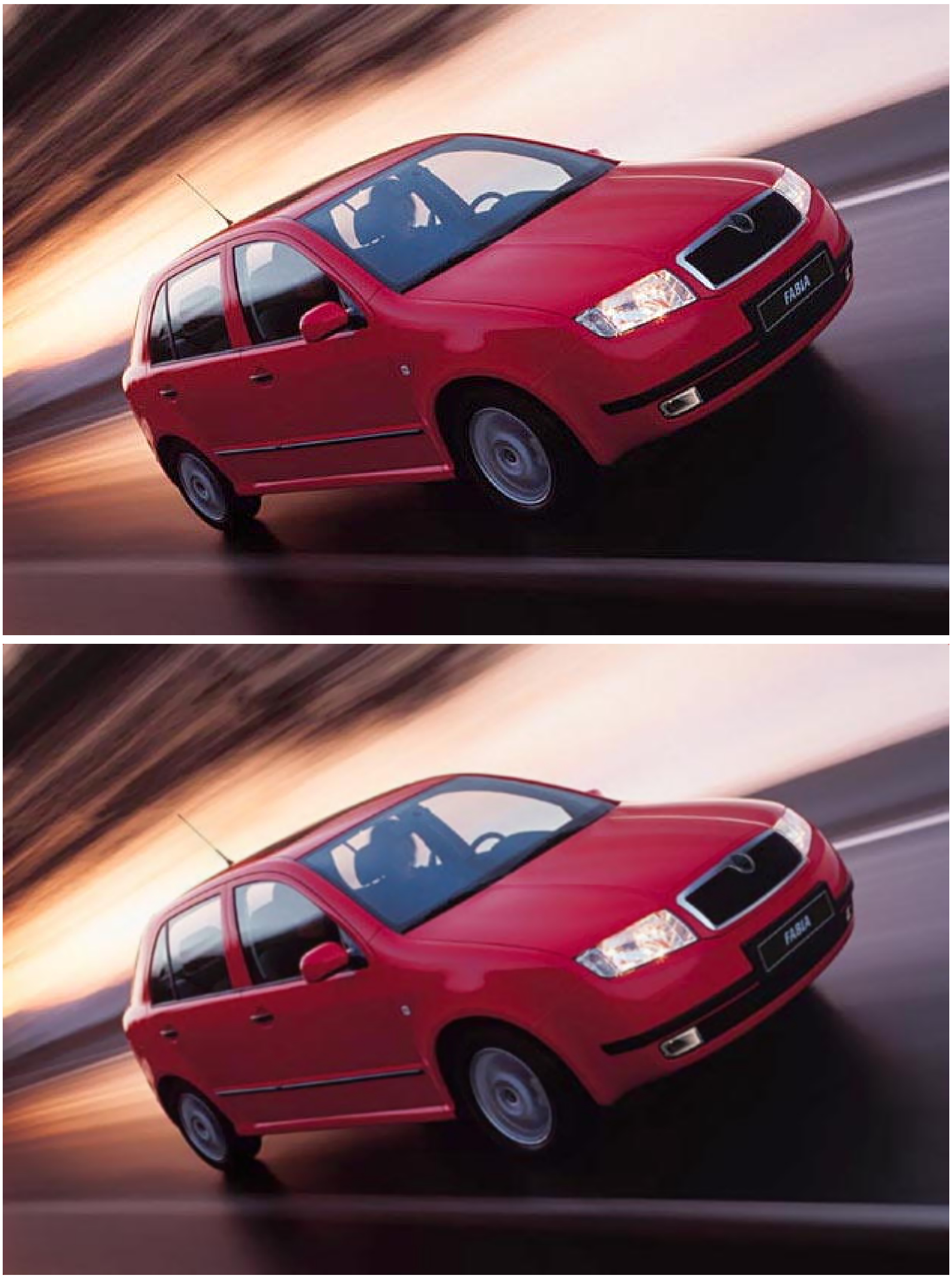
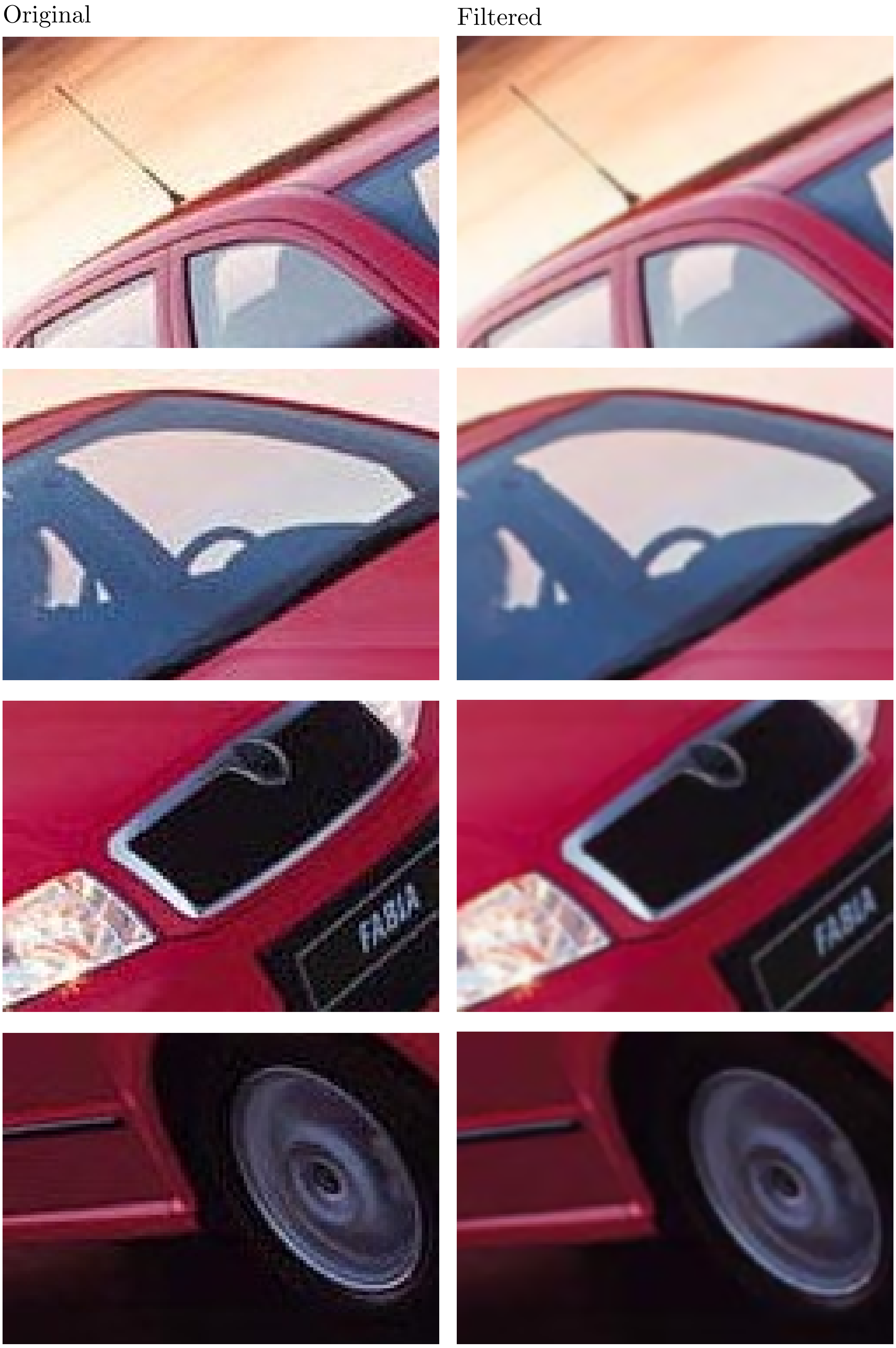


Figure 5.9: An example of JPEG restoration. Top: a 630×420 pixel jpeg image (40kB) as available on the internet (courtesy Škoda auto). Bottom: resampled to 888×592 and filtered with anisotropic NL diffusion, $T = 3$, $\tau = 1$, $p = 0.7$, $\varphi_2 = 1$. JPEG artefacts and aliasing are substantially reduced at the price of a slight softening of the image.



Original

Filtered

Figure 5.10: Details from Fig. 5.9: original on the left, filtered on the right.

5.2 Monotonicity enhancement

In this section, we test the performance of the monotonicity-enhancing NL diffusion as developed in Chapter 4. The experiments include some artificial examples, grey-level images, and range data for 3D reconstruction.

5.2.1 1D example

The results of several methods of nonlinear diffusion applied to a simple 1D function are shown for comparison in Figure 5.11. In the center row, the preference of the classical nonlinear diffusion for (piecewise) constant functions can be seen: depending on the parameters, it either approximates the input by several steps, or bends the function near the extrema. At the bottom, two results of the nonlinear diffusion of first derivatives are shown for comparison: the left one, where the diffusion was controlled by first derivatives, shares some properties of the ordinary diffusion (rounding of the corner) but behaves better near the ends of continuous segments. The right one, where the diffusion was controlled by a function of second derivatives, is also able (for carefully chosen parameters) to precisely locate the corner in the function values, i.e. the discontinuity of the second derivative.

5.2.2 Artificial 2D data

An experiment with synthetic noisy 2D data is presented in Figure 5.12. The image gradient is larger on the sloped surface; to remove the noise from that area with the ordinary nonlinear diffusion, a larger diffusivity parameter λ has to be chosen, which leads to a higher risk of blurring the discontinuities. You can observe this phenomenon, as well as the bending of increasing segments, in Figure 5.12c. In contrast, our method controlled by second derivatives according to equation (4.11) behaves in the same way regardless of the surface slope, the discontinuity is well preserved for a large interval of parameters, and the function bending is avoided. These properties are demonstrated in Figure 5.12d. Please note that figures 5.12c and 5.12d show the best filtering result (evaluated using the MAD distance form the ideal, noise-free data) of each method. Still, you may notice that some noise has remained near the areas of higher second derivatives, i.e. near the discontinuities and corners, where the diffusion is inhibited. This misbehaviour can be eliminated by anisotropic filter where the diffusion is controlled by a diffusion tensor; the result of two anisotropic filters can be seen in the bottom row of Figure 5.12 (the classical one on the left and the derivative-smoothing on the right). The diffusion parameters λ and T were chosen autonomously in these two examples (using the Black–Sapiro estimation of λ, and the decorrelation criterion (3.3) for the stopping time); this approach leads to a rather rough result in the classical case 5.12e, but yields nice and well filtered data using the anisotropic monotonicity-enhancing procedure.

If we evaluate the four filtering results of Figure 5.12 using the MAD distance of the filtered result from the ideal data $\widetilde{\mathbf{f}}$, we obtain the measurements shown in Table 5.2. You can see that the two isotropic filters (first two lines of Table 5.2) yield similar numbers. However, not all kinds of errors can be considered equivalent. We

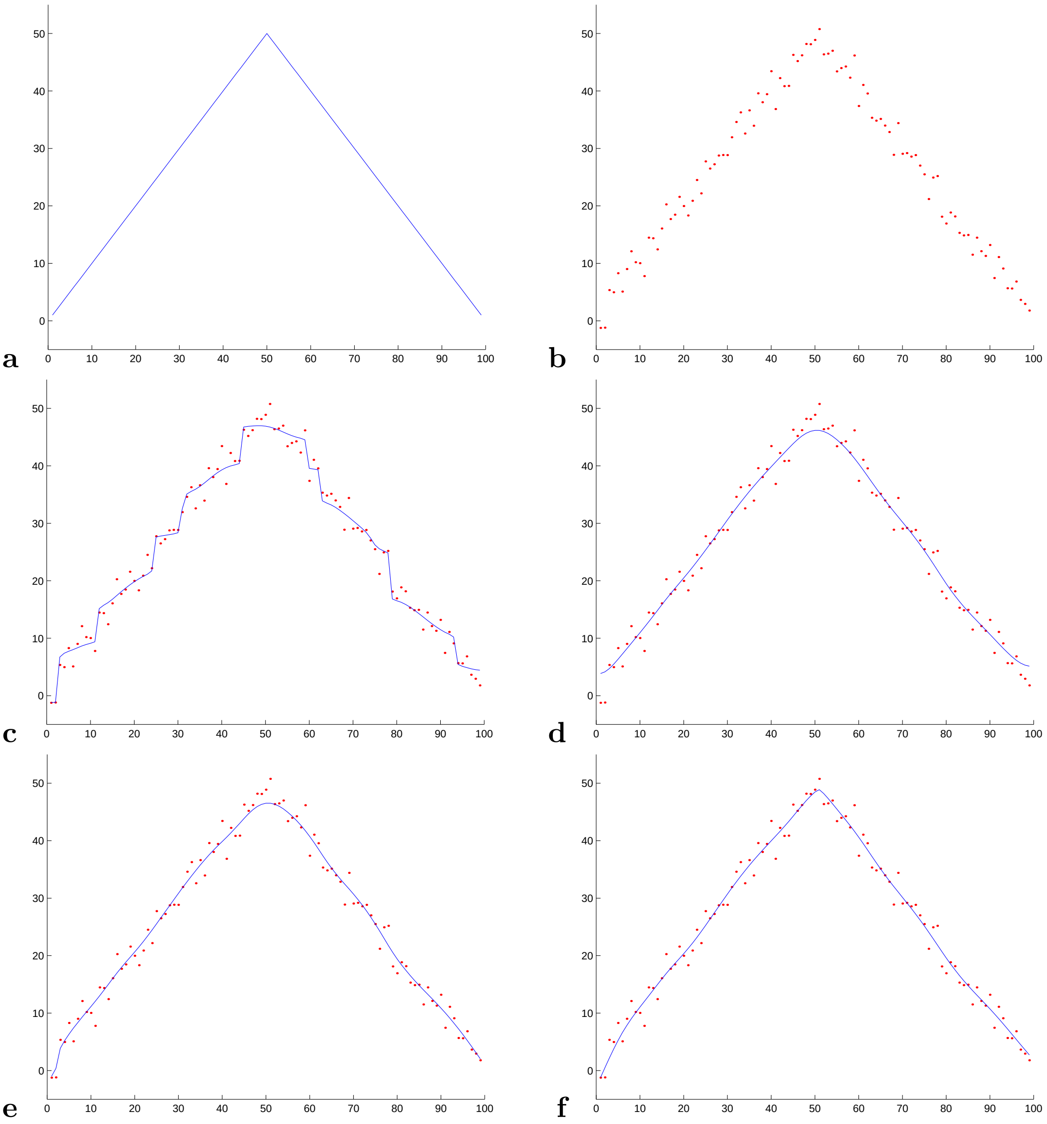


Figure 5.11: Gaussian noise was added to a triangle function **(a)** to obtain the noisy data **(b)**, used to test the diffusion filtering methods. The filtering results (continuous lines) are shown together with the noisy input (dots).
(c) Ordinary nonlinear diffusion, $\sigma = 1, \lambda = 1, T = 10$.
(d) Ordinary nonlinear diffusion, $\sigma = 1, \lambda = 3, T = 10$.
(e) Filtering by nonlinear diffusion of first derivatives, $\sigma = 1, \lambda = 1, T = 10$. The diffusion is controlled by first derivatives according to (4.12).
(f) Filtering by nonlinear diffusion of first derivatives, $\sigma = 3, \lambda = 0.05, T = 10$. The diffusion is controlled by the second order derivatives as described by (4.11).

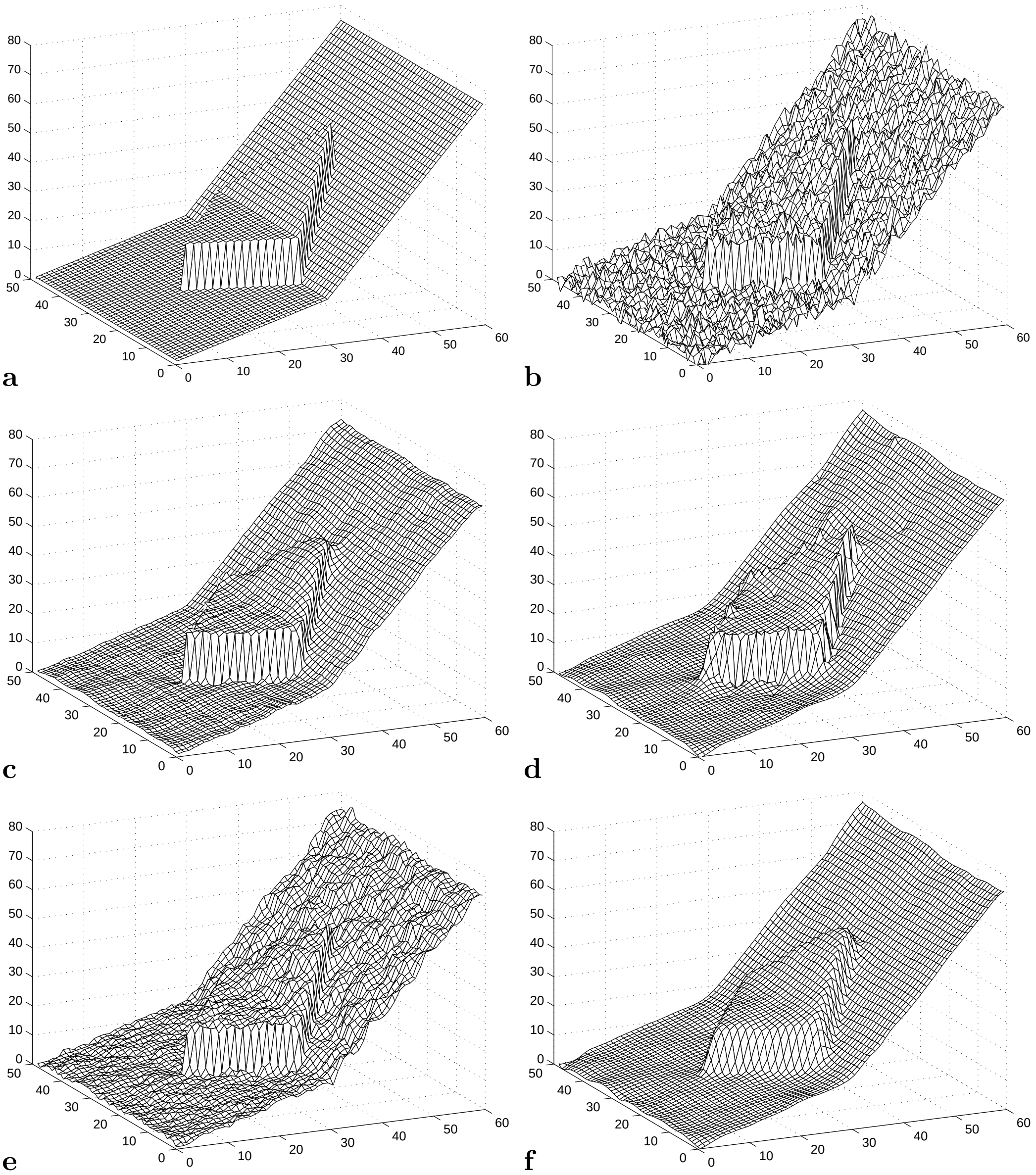


Figure 5.12: Experiment with two-dimensional data. **(a)** Ideal data. **(b)** Noisy input.
(c) Optimal result using classical isotropic nonlinear diffusion of (b), $T = 3$.
(d) Optimal result when the noisy input was filtered with the isotropic nonlinear diffusion of first derivatives, controlled by second derivatives, $T = 6$. In (c)–(d), the parameters $\sigma = 1, \tau = 1$ were employed, and the diffusivity parameter λ was estimated by the Perona–Malik procedure from $p = 0.9$ in each step.
(e) Classical anisotropic NL diffusion resulting in $T = 4$.
(f) Anisotropic NL diffusion for monotonicity enhancement leads to $T = 10$. In (e)–(f), we used the Black–Sapiro estimation of λ, and the stopping time was determined autonomously using the decorrelation criterion.

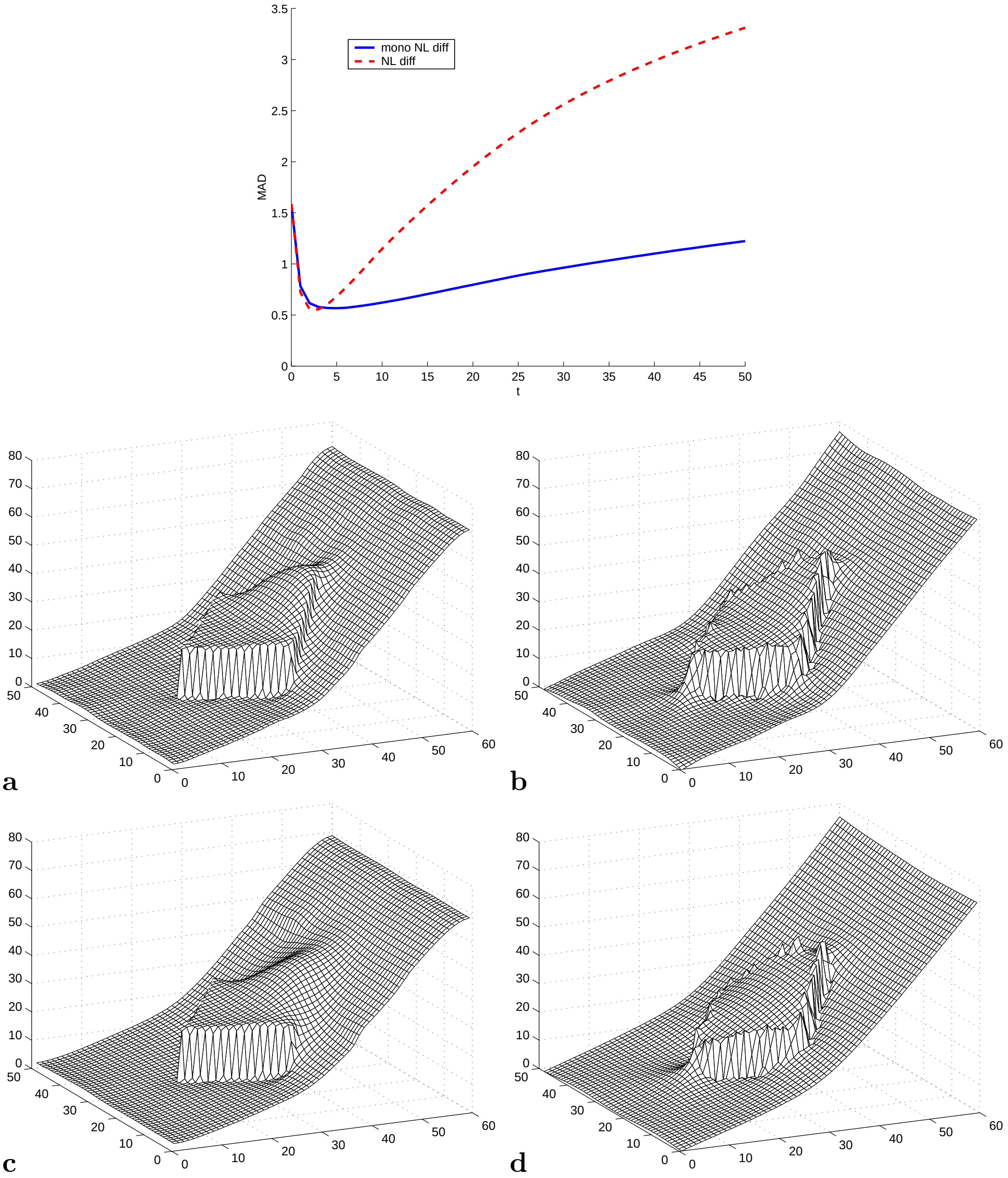


Figure 5.13: Top: a graph showing how the MAD distance of the filtered image from the ideal data (fig. 5.12) develops with the time for the classical, and the monotonicity-enhancing isotropic diffusion. Below, some results of the classical diffusion (left) and the diffusion on the derivatives (right) are shown for comparison. Middle row: the result of twice the optimal time, i.e. $T = 6$ for the classical, and $T = 10$ for the monotonicity-enhancing diffusion. Bottom: the results for which $\text{MAD}(\mathbf{u}(T), \widetilde{\mathbf{f}}) \approx 1.2$; left: classical NL diffusion, $T = 12$, right: NL diffusion of derivatives, $T = 50$.

Figure	method	T	$\mathrm{MAD}(\mathbf{u}(T), \widetilde{\mathbf{f}})$	$\mathrm{corr}(\mathbf{u}(0) - \mathbf{u}(T), \mathbf{u}(T))$
5.12b	—	0	1.58	—
5.12c	isotropic, classical	3	0.555	0.1213
5.12d	isotropic, derivatives	5	0.567	0.0357
5.12e	anisotropic, classical	4	0.764	0.0716
5.12f	anisotropic, derivatives	10	0.560	0.0527

Table 5.2: Quantitative evaluation of the filtering results shown in Figure 5.12

have seen in Figure 5.12 that the monotonicity-enhancing diffusion gives visually better results on piecewise monotone data.

Figure 5.13 demonstrates another fact: choosing the model which corresponds better to the particular data properties leaves us more freedom for the intricate task of stopping time selection. Figure 5.13 top plots the MAD quality of the filtered output against the diffusion time of the two methods: the monotone model on monotone data leads to a stable result for a wide interval of time T. The center row of Figure 5.13 shows how different the results of the diffusion become if we make a mistake in the choice of T. The bottom row then visualises that the same MAD distance from the ideal data does not necessarily mean an equivalent result; with the classical NL diffusion, the discontinuity of the data on the sloped surface is completely lost.

These observations on time stability of the solution are confirmed by the last two lines of Table 5.2 and the bottom row of Figure 5.12. In these anisotropic examples, the stopping time T was determined autonomously by the decorrelation criterion 3.3. While resulting in less-than-optimal solution of the classical diffusion filter, the more appropriate monotonicity-enhancing method gives an excellent output, outperforming other filters both visually and quantitatively.

5.2.3 Range data

We present an example of range data filtering, one of the main motivations for the development of monotonicity-enhancing NL diffusion filter. The input range data are shown in Figure 5.14; the image pixels represent distance from the measurement device (brighter images are closer). Although this grey-level image does not look noisy, small-scale fluctuations are present. The noise becomes clearly visible if we visualise the data as a 3D mesh (see detail in Fig. 5.15 left) or render a 3D look of the surface (Fig. 5.16 top).

Figures 5.15 right and 5.16 bottom show the same data after five iterations of anisotropic monotonicity-enhancing NL diffusion. Most of the noise has been eliminated, the images appear smoother, but the important discontinuities and also the corners (e.g. at the table leg) have been preserved. Note that both images in Fig. 5.16 were rendered using the same illumination conditions and other properties; the rough surface of the original data reflects only a small portion of the incoming light. Unfortunately, as the ground truth is not available, we cannot provide any

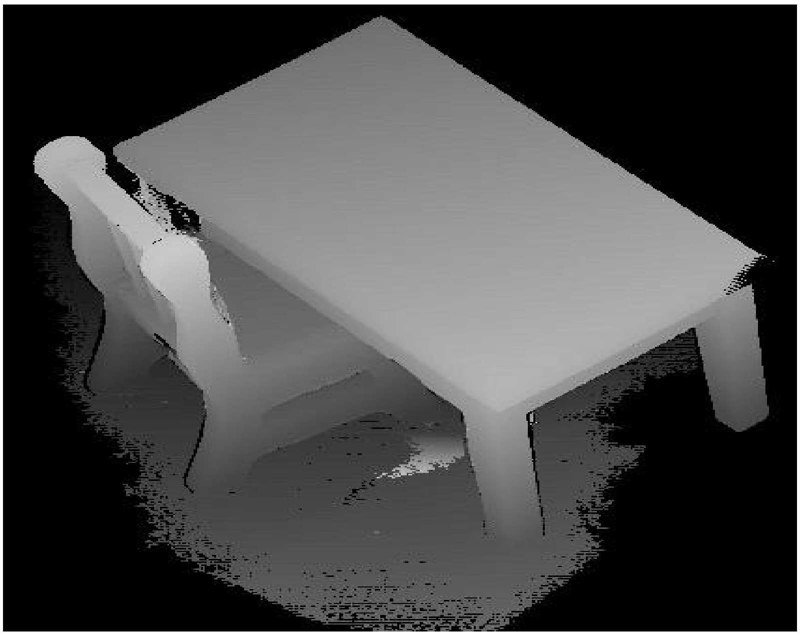


Figure 5.14: Input range data (from `http://eewww.eng.ohio-state.edu/~flynn/3DDB/RID/`).

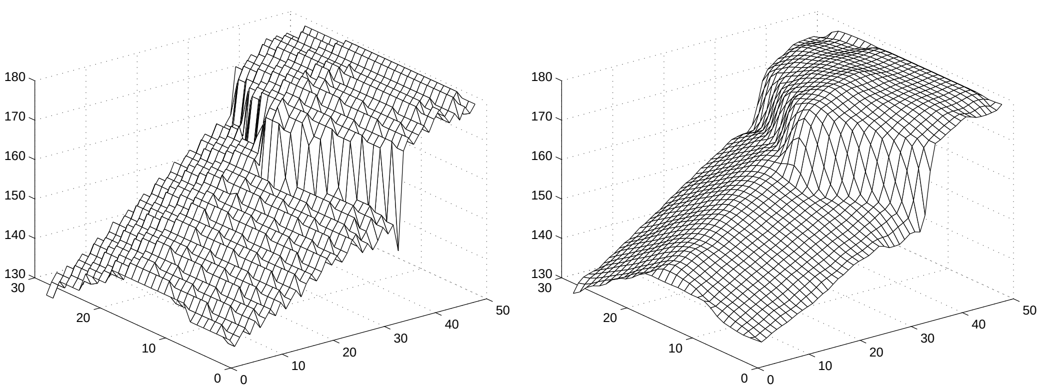
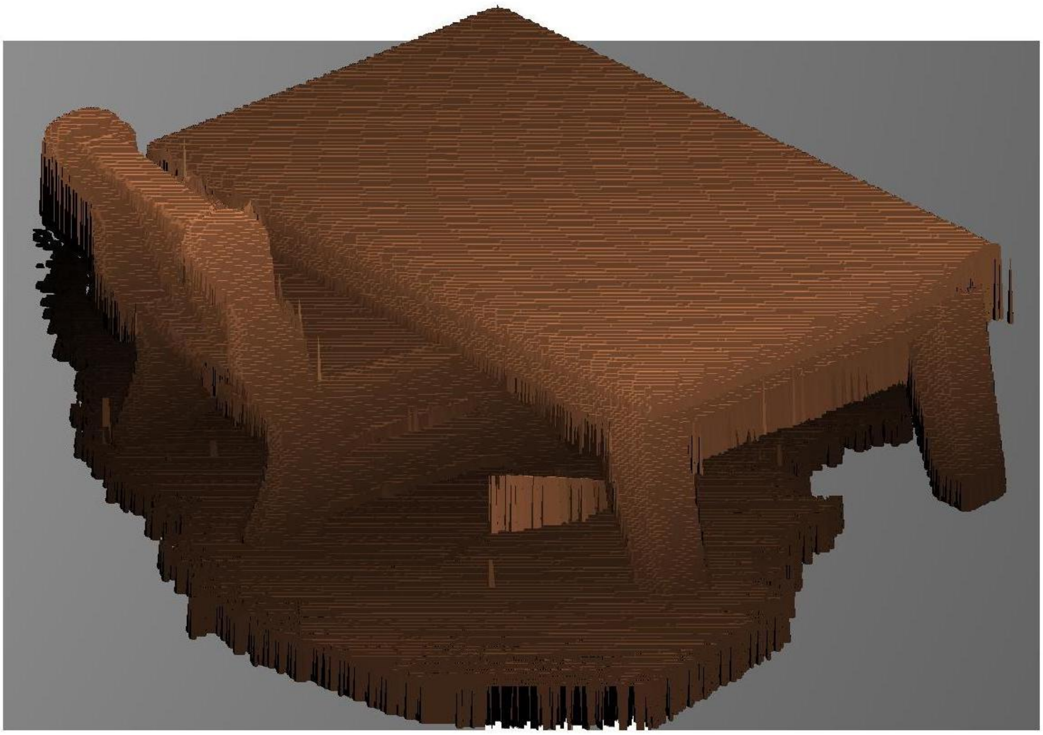


Figure 5.15: A detail from the range data (corner of the table) visualised as a 3D mesh. Left: original data. Right: filtered using anisotropic monotonicity-enhancing NL diffusion (as in Fig. 5.16 bottom).



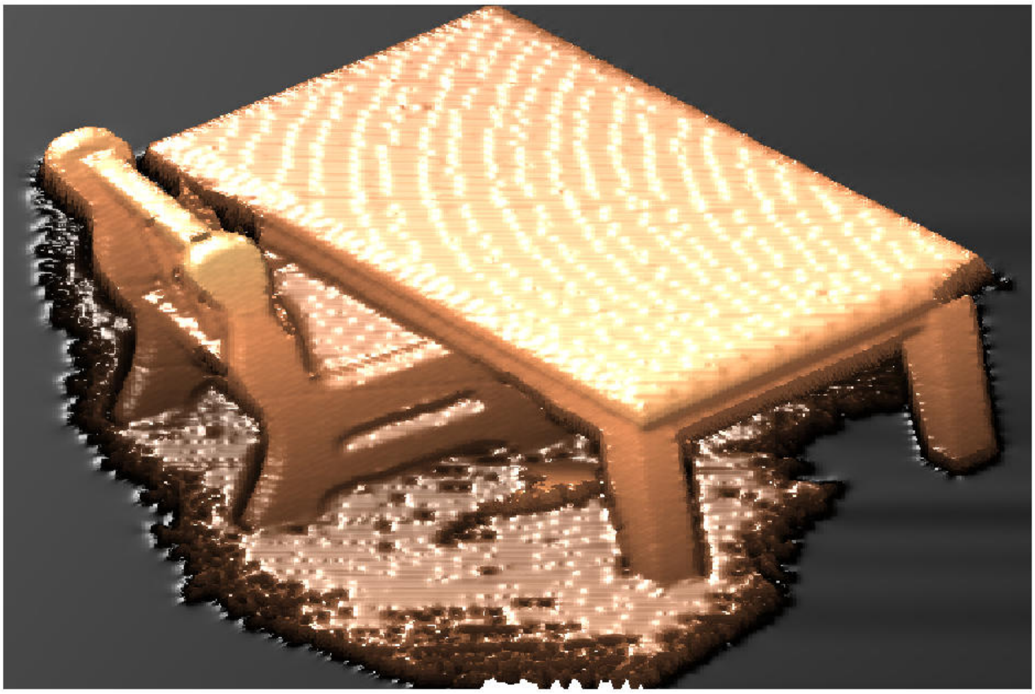


Figure 5.16: Range data filtering. Top: original depth data from Fig. 5.14 rendered into a 3D look using MATLAB graphics. Bottom: filtered by anisotropic monotonicity-enhancing diffusion, $T = 5$, $\tau = 1$, $p = 0$, $\varphi_2 = 0.2$. Both images were rendered with the same illumination conditions and camera parameters.

assesment of the filtering quality better than this visual inspection.

In this experiment, we applied the filtering procedure to the range data in the way usual for normal images. However, range data typically contain holes, some parts of the measurements are either missing or unreliable. To make NL diffusion more suitable to range data filtering, these cases should be treated systematically and we should prevent some parts of the image from influencing the result.

5.2.4 Image filtering

Although not designed originally for that type of data, the monotonicity-enhancing filter may be applied successfully to photographic images. Figure 5.17 top gives an example of an image consisting mainly of regions of slow, gradual transitions from dark to light colors (cheek, nose) and discontinuities or edges between regions, so piecewise monotonicity can be assumed. The results of the classical nonlinear diffusion from equation (2.71) and the derivative-based diffusion are shown for comparison in the middle row of Figure 5.17; both methods perform well in removing the noise, the diffusion using first derivatives is better at preserving the gradual transitions between different light intensities. At the bottom, in the plots of one line extracted from the image, you can observe in more detail how the function values, slopes and discontinuities develop with the diffusion time of both methods. Rather for interest than for a practical application, Figure 5.18 shows the image at several higher times of both diffusion methods.

5.3 Final remarks

To summarize the experiments, we make two small remarks. First, we have praised the monotonicity-enhancing diffusion of directional derivatives for its superior filtering performance on piecewise monotone data. We should stress, however, that this success cannot be extended to any kind of data. Obviously enough, the piecewise linear model hidden behind the procedure is more general (less restrictive) than the piecewise constant one, and is therefore weaker at filtering of images which should be piecewise constant.

A similar note can be made about the performance of the autonomous decorrelation stopping time estimation 3.4. We observed that the best results are obtained if, to data with some properties, we apply a filter respecting those properties: filter piecewise constant data with classical edge-enhancing diffusion ; run monotonicity-enhancing procedure on piecewise linear data; etc.

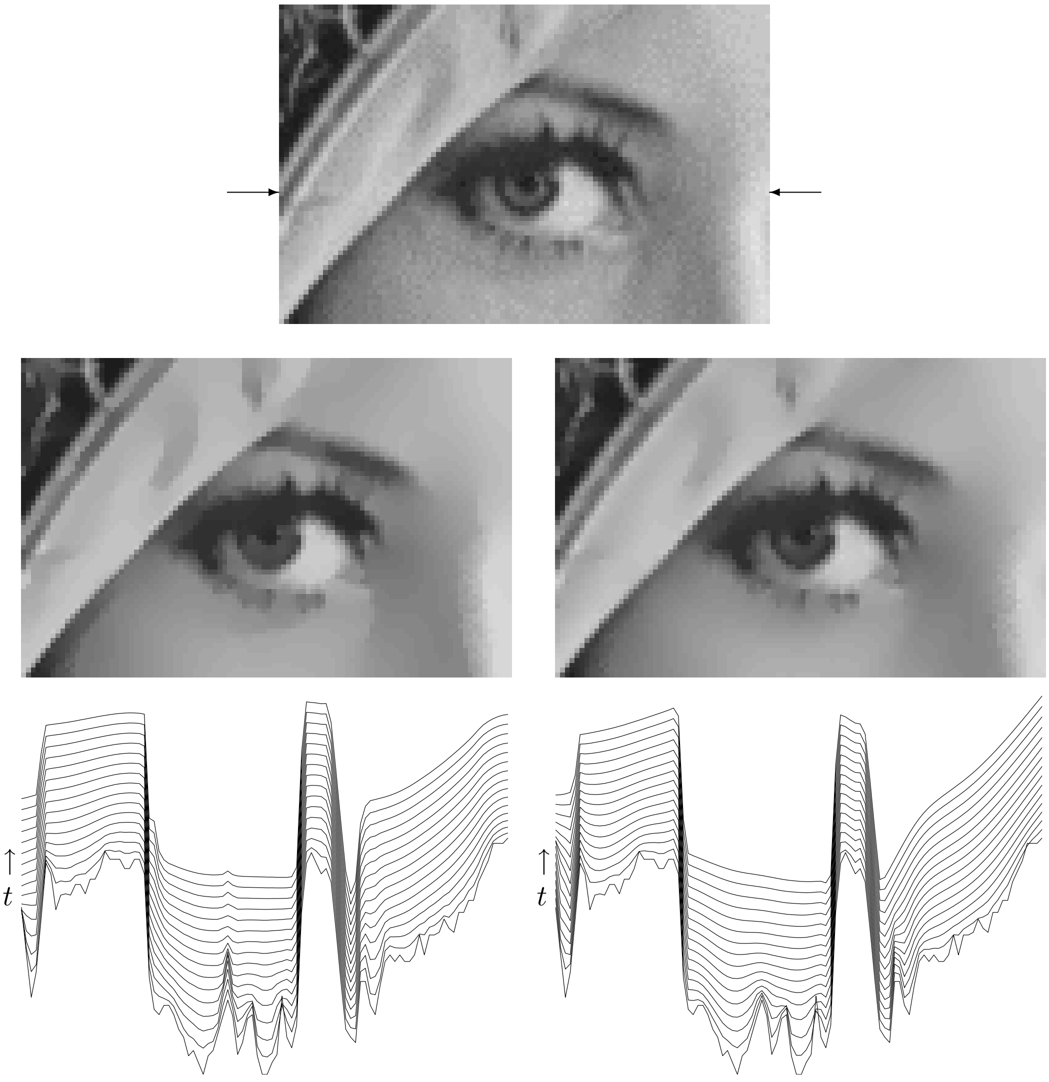


Figure 5.17: Top: Input image, a detail of Lena.
Center left: ordinary nonlinear diffusion of the input image; center right: the same using nonlinear diffusion of first partial derivatives. Both methods were run with the parameters $\sigma = 1$, $\lambda = 1$, $\tau = 1$, $T = 5$.
Bottom: one horizontal line in the image (passing through the eye) develops with time until $T = 30$. Note that the continuous function segments diffused by the classical method on the left approach a (piecewise) constant function; the nonlinear diffusion of first partial derivatives (on the right) tends to a function piecewise linear.
These images can also be seen at
http://cmp.felk.cvut.cz/˜mrazekp/Diffusion/diffusion.html.

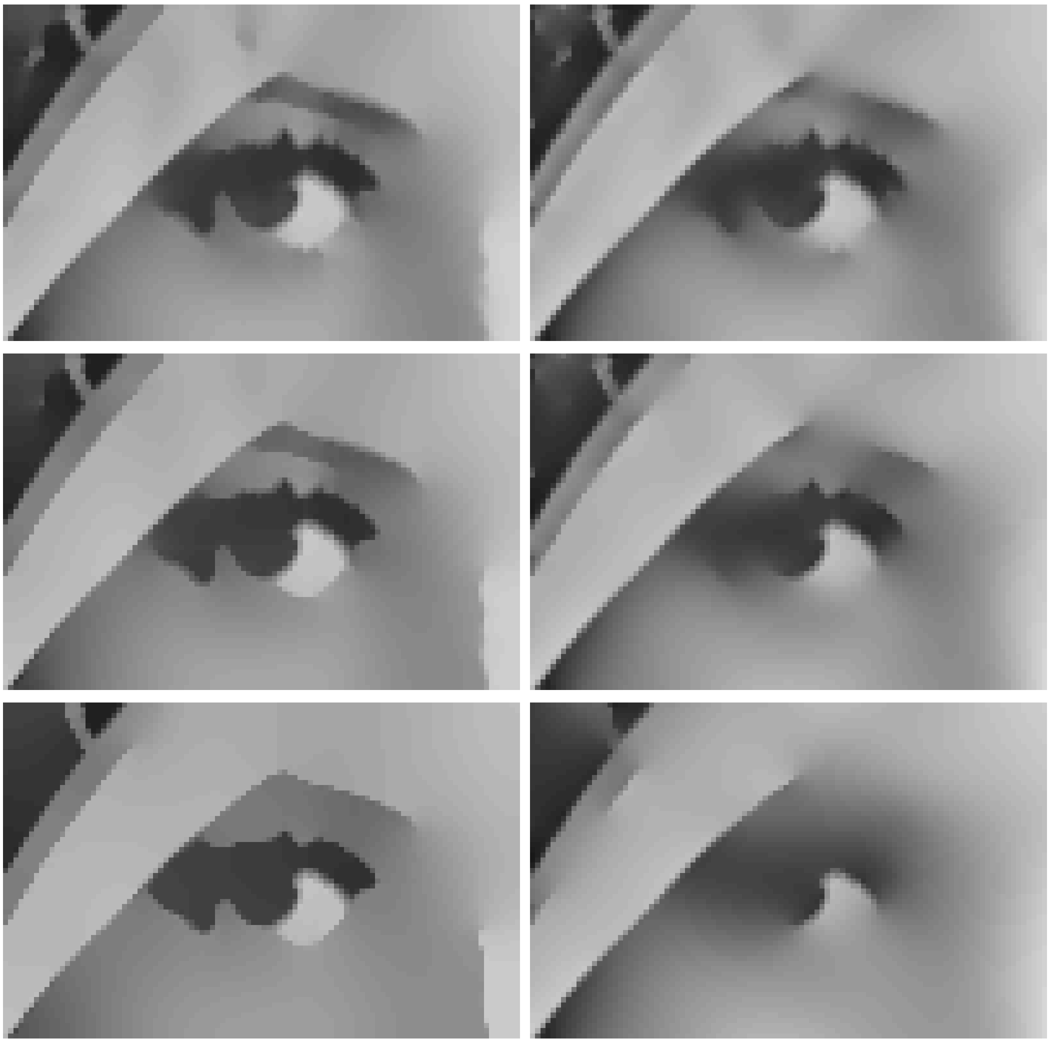


Figure 5.18: The section of Lena image diffused more by the classical isotropic NL diffusion (left) and the NL diffusion of image derivatives (right); the time instants are (top to bottom) $T = 10$, $T = 20$, and $T = 50$, respectively.

6 Conclusion

Motivated by the need to filter additive noise from discrete data expected to be piecewise continuous and piecewise monotone (i.e. piecewise increasing or decreasing on continuous regions), we started this text by reviewing previous work on nonlinear diffusion, a powerful image restoration technique; we discussed its properties, and presented numerical methods for isotropic and anisotropic cases of the filter. Nonlinear diffusion creates iteratively simplified versions of the input and tends to piecewise constant functions.

As main contribution of this thesis, we extended NL diffusion for the special case where the trends in the data are of importance. This generalization of nonlinear diffusion uses a piecewise linear model in the equations instead of the piecewise constant one. The task is accomplished by nonlinear vector-valued diffusion of partial derivatives of image data combined with restoration of necessary properties of the directional derivatives. We develop the algorithms for monotonicity enhancement (isotropic and anisotropic), and analyse the theoretical properties of the new approach in Chapter 4; experimental results appear in Section 5.2. The method is applicable to filtering or smoothing of sampled functions expected to be piecewise continuous, piecewise monotone or piecewise linear. Many images fulfill these properties, range data for 3D reconstruction represent a particular example.

On the pursuit of this main contribution, we ran across several open questions, and solved some of them. So in Section 3.1 we develop a novel method to estimate the optimal stopping time for iterative image restoration techniques such as nonlinear diffusion. The stopping time is chosen so that the correlation of signal and noise is minimised. The new criterion outperforms other time selection strategies and yields near-optimal results for a wide range of noise levels and for various filtering methods. The decorrelation criterion is also more general than previous approaches, being based only on the assumption that the noise and the signal in the input image are uncorrelated; no knowledge on the variance of the noise, and no training images are needed to tune any parameters of the method.

We have analysed the possibilities for consistent positive directional splitting of anisotropic diffusion on a 3×3 window in Section 3.2; such a splitting is needed to transform the continuous anisotropic diffusion equations into efficient numerical algorithms. We have shown that consistent positive splitting exists if the interval $[b, min(a,c)]$, formed from the elements of the diffusion tensor $\boldsymbol{D} = \left(\begin{smallmatrix} a & b \\ b & c \end{smallmatrix}\right)$, is nonempty. We have also derived the formulas for the directional diffusivities depending on a single diffusivity parameter, and demonstrated experimentally that the directional splitting reveals better properties (regarding e.g. rotational symmetry and sensitivity to the time step size) if the splitting parameter is chosen from the interior of the admissible interval.

As a contribution of minor theoretical interest but important in practice, we have summarized various aspects of the diffusion filtering, including previous ideas on the choice of the diffusion parameters (and adding some of ourselves), to create an autonomous image-filtering procedure in Section 3.3. The performance of this algorithm is demonstrated experimentally in Chapter 5.

Future work. These are some of possible directions we can see now for further development of the filtering methods:

- In the repeatedly mentioned task of 3D reconstruction from visual information, some parts of the objects may be either invisible by the camera(s), occluded from the light source, or its measurements made otherwise unavailable. To make our filter even more applicable, it would be desirable to handle these cases of missing data systematically.

- On a similar problem: with many experimental methods, some parts of the measured data can be considered more reliable than others. If such an information is available, it should be employed for the processing. We would suggest to incorporate some kind of data reliability measure, for example in the form of weights associated with the individual data, into the model. (In this context, the missing data of the previous paragraph can be understood as data of zero reliability.)

- Diffusion filtering can be applied in many areas we have not explored yet; medical imaging represents a classical example, JPEG restoration has been mentioned only briefly. Where motivated by the data properties, the monotonicity-enhancing procedure should be employed and its performance evaluated.

- If needed for an application, the monotonicity-enhancing procedure could be extended to three (or more) dimensions. Similarly to 2D, the main difficulty would consist in the design of a derivative-correcting algorithm.

Notation

Ω (continuous) image domain, typically a rectangular area $[0, x_{\max}] \times [0, y_{\max}] \subset \mathbb{R}^2$

$\partial\Omega$ boundary of the image domain Ω

$|\Omega|$ measure of Ω

$\overline{\Omega}$ closure of Ω

$\mathbf{f}$ (also $\mathbf{u}, \mathbf{v}, \ldots$) (image) data, either continuous, $\mathbf{f}\colon \Omega \to \mathbb{R}$, or discrete, stored in a vector $\mathbf{f} = (f_i)_{i=1,\ldots,N}$ where the sample f_i represents the value $\mathbf{f}(\vec{x}_i)$, $\vec{x}_i \in \Omega$. For 2D digital images, $\mathbf{f} = (f_{i,j})_{\substack{i=1,\ldots,N_i \\ j=1,\ldots,N_j}}$.

$\widetilde{\mathbf{f}}$ ideal, noise-free data

$h_{i,j}$ sampling kernel for position i, j

n noise, or normal to region boundary

N_i, N_j number of discrete samples in the directions of axes x and y, respectively

Δx, Δy discretization grid size; we usually set $\Delta x = \Delta y = 1$

t scale parameter, or time of the diffusion process

$\partial_t \mathbf{u}$ shorthand notation for the partial derivative $\frac{\partial \mathbf{u}}{\partial t}$

$\langle x, y \rangle$ scalar product (or dot product) of vectors x and y

∇ gradient operator; for data $\mathbf{u} \in \mathbb{R}^N$, ∇ is a column vector of N elements, $\nabla \equiv \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_N}\right)^T$ where x_i, $i = 1, \ldots, N$ are the coordinate axes. For the case of normal images, $N = 2$ and the image gradient is $\nabla \mathbf{u} = \left(\frac{\partial \mathbf{u}}{\partial x}, \frac{\partial \mathbf{u}}{\partial y}\right)^T$.

Δ Laplacian operator, $\Delta \mathbf{u} \equiv \nabla^2 \mathbf{u} = \langle \nabla, \nabla \mathbf{u} \rangle = \left(\frac{\partial^2 \mathbf{u}}{\partial x^2} + \frac{\partial^2 \mathbf{u}}{\partial y^2}\right)$ where the last transformation assumes that $\mathbf{u}$ is defined on $\mathbb{R}^2$

div divergence operator, $\operatorname{div} \psi = \langle \nabla, \psi \rangle$

G_σ	Gaussian kernel; in $\mathbb{R}^N$, $G_\sigma(x) = \frac{1}{(2\pi\sigma^2)^{N/2}}\ e^{-x^T x/2\sigma^2}$
ψ	flux
$\boldsymbol{D}$	diffusion tensor
g	diffusivity function
λ	parameter of the diffusivity function
p	percentile for the Perona-Malik estimation of λ, or parameter for directional splitting of anisotropic diffusion
$\mathbf{u}_\sigma$	image smoothed by convolution with a Gaussian kernel, $\mathbf{u}_\sigma = G_\sigma * \mathbf{u}$
$\nabla\mathbf{u}_\sigma$	gradient of the smoothed image, $\nabla\mathbf{u}_\sigma = (\nabla G_\sigma) * \mathbf{u} = \nabla(G_\sigma * \mathbf{u})$
$\boldsymbol{I}$	identity matrix
$\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{Q}$	matrices containing the diffusivity information
σ	noise scale, width of the Gaussian kernel G_σ
T	stopping time of the diffusion
$\boldsymbol{S}_\varrho$	structure tensor
ϱ	integration scale
$\varphi_{1,2}$	eigenvalues of the diffusion tensor
$\mu_{1,2}$	eigenvalues of the structure tensor
$\vec{v}_{1,2}$	eigenvectors of the structure and diffusion tensors
$\mathrm{diag}(a,b)$	diagonal matrix $\left(\begin{smallmatrix} a & 0 \\ 0 & b \end{smallmatrix}\right)$
τ	iteration time step
$\mathbf{u}^k$	solution of the diffusion process at time $t_k = k \cdot \tau$
$\mathcal{N}(i)$	set of the neighbours of pixel i
g_{ij}	diffusivity belonging to the connection between pixels i and j
$\overline{x}, E(x)$	mean (or expectation) of a vector $x \in \mathbb{R}^N$: $\overline{x} = E(x) = \frac{1}{N}\sum_{i=1}^N x_i$
$\mathrm{var}(x)$	variance of x, $\mathrm{var}(x) = E\big[(x - \overline{x})^2\big]$
$\mathrm{cov}(x,y)$	covariance of two vectors, $\mathrm{cov}(x,y) = E\left[(x - \overline{x}) \cdot (y - \overline{y})\right]$
$\mathrm{corr}(x,y)$	correlation coefficient, $\mathrm{corr}(x,y) = \frac{\mathrm{cov}(x,y)}{\sqrt{\mathrm{var}(x) \cdot \mathrm{var}(y)}}$

SNR	signal-to-noise ratio, SNR $= \frac{\mathrm{var}\left(\tilde{\mathbf{f}}\right)}{\mathrm{var}(n)}$
β_i	direction of 1D diffusion process (for directional splitting of anisotropic diffusion)
α_i	diffusivity in the direction β_i
K	number of connected regions Ω_k needed to partition Ω so that $\mathbf{u}$ is monotone on each Ω_k
$\mathbf{v}, \mathbf{w}$	fields of directional derivatives for the monotonicity-enhancing NL diffusion
$\tilde{\mathbf{v}}, \tilde{\mathbf{w}}$	$\mathbf{v}, \mathbf{w}$ smoothed by convolution with G_σ
$\hat{\mathbf{z}}$	result of the diffusion of derivatives, $\hat{\mathbf{z}} = [\mathbf{v}^k, \mathbf{w}^k]^T$
$\boldsymbol{C}$	matrix expressing the discrete gradient field constraint
$\mathbf{z}$	orthogonal projection of $\hat{\mathbf{z}}$ into the null space of $\boldsymbol{C}$, i.e. directional derivatives after enforcement of the gradient field constraint
$e_{i,j}$	error of the gradient field constraint at position i, j
u_0	integration constant for the reconstruction of $\mathbf{u}$ from $\mathbf{v}, \mathbf{w}$

Bibliography

[1] S. Acton and A. Bovik. Piecewise and local image models for regularized image restoration using cross-validation. *IEEE Transactions on Image Processing*, 8(5):652–665, 1999.

[2] S. T. Acton and A. C. Bovik. Nonlinear image estimation using piecewise and local image models. *IEEE Transactions on Image Processing*, 7(7):979–991, July 1998.

[3] L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel. Axioms and fundamental equations of image-processing. *Archive For Rational Mechanics And Analysis*, 123(3):199–257, 1993.

[4] E. Bänsch and K. Mikula. A coarsening finite element strategy in image selective smoothing. *Computing and Visualisation in Science*, pages 53–61, 1997.

[5] M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging.* Institute of Physics Publishing, Bristol, Great Britain, 1998.

[6] M. J. Black and G. Sapiro. Edges as outliers: anisotropic smoothing using local image statistics. In M. Nielsen, P. Johansen, O. F. Olsen, and J. Weickert, editors, *Scale-Space Theories in Computer Vision*, number 1682 in Lecture Notes in Computer Science, pages 259–270. Springer, Berlin, Germany, 1999.

[7] M. J. Black, G. Sapiro, D. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 7(3):421–432, March 1998.

[8] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, 1987.

[9] D. Caspi, N. Kiryati, and J. Shamir. Range imaging with adaptive color structured light. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):470–480, May 1998.

[10] F. Catté, P.-L. Lions, J.-M. Morel, and T. Coll. Image selective smoothing and edge-detection by nonlinear diffusion. *SIAM Journal on Numerical Analysis*, 29(1):182–193, 1992.

[11] A. Chambolle and P.-L. Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76:167–188, 1997.

[12] I. C. Dolcetta and R. Ferretti. Optimal stopping time formulation of adaptive image filtering. *Applied Mathematics and Optimization*, 2000. To appear.
Available at `http://www.mat.uniroma1.it/ricerca/TMR`.

[13] M. Haindl. Recursive square-root filters. In A. Sanfeliu, J. J. Villanueva, M. Vanrell, R. Alquezar, A. K. Jain, and J. Kittler, editors, *Proceedings of the 15th IAPR Int. Conf. on Pattern Recognition*, volume II, pages 1018–1021, Los Alamitos, USA, September 2000. IEEE Computer Society Press.

[14] A. Handlovičová, K. Mikula, and F. Sgallari. Variational numerical methods for solving nonlinear diffusion equations arising in image processing. *Journal of Visual Communication and Image Representation*, 2001. To appear.

[15] J. J. Koenderink. The structure of images. *Biological Cybernetics*, (50):363–370, 1984.

[16] Z. Krivá and K. Mikula. An adaptive finite volume method in processing of color images. In *Proceedings of Algoritmy 2000, Conference on Scientific Computing*, pages 174–187, September 2000.

[17] T. Lindeberg. *Scale-Space Theory in Computer Vision.* Kluwer Academic Publishers, 1994.

[18] D. G. Luenberger. *Optimization by vector space methods.* John Wiley and Sons, 1969.

[19] K. Mikula and N. Ramarosy. Semi-implicit finite volume scheme for solving nonlinear diffusion equations in image processing. *Numerische Mathematik*, 2001. To appear.

[20] P. Mrázek. Enhancing monotonicity by nonlinear diffusion of image derivatives. In T. Svoboda, editor, *Czech Pattern Recognition Workshop 2000*, pages 21–26, Peršlák, Czech Republic, Feb 2000. Czech Pattern Recognition Society.
Available at `ftp://cmp.felk.cvut.cz/pub/cmp/articles/mrazek/Mrazek-cprw2000.ps.gz`.

[21] P. Mrázek. Monotonicity enhancing nonlinear diffusion. *Journal of Visual Communication and Image Representation*, 2001. To appear.

[22] P. Mrázek. Selection of optimal stopping time for nonlinear diffusion filtering. In *IEEE Workshop on Scale-Space and Morphology in Computer Vision.* Springer-Verlag, July 2001. To appear.

[23] P. Mrázek and M. Navara. Consistent positive directional splitting of anisotropic diffusion. In B. Likar, editor, *Computer Vision Winter Workshop 2001*, pages 37–48, Ljubljana, Slovenia, Feb 2001. Slovenian Pattern Recorgnition Society.
Available at `ftp://cmp.felk.cvut.cz/pub/cmp/articles/mrazek/Mrazek-cvww01.pdf`.

[24] 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.

[25] K. N. Nordström. Biased anisotropic diffusion: a unified regularization and diffusion approach to edge detection. *Image and Vision Computing*, 8(4):318–327, Nov 1990.

[26] T. Pajdla. Laser plane range finder - the implementation at the CVL. Technical Report Nr. K335-95-98, Czech Technical University, Prague, Oct. 1995.
Available at `ftp://cmp.felk.cvut.cz/pub/cmp/articles/pajdla/lprf.ps.Z`.

[27] A. Papoulis. *Probability and Statistics.* Prentice-Hall, 1990.

[28] P. Perona and J. Malik. Scale-space and edge-detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[29] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipies in C. The Art of Scientific Computing.* Cambridge University Press, 2-nd edition, 1992.

[30] T. Preußer and M. Rumpf. An adaptive finite element method for large scale image processing. *Journal of Visual Communication and Image Representation*, 2001. To appear.

[31] E. Radmoser, O. Scherzer, and J. Weickert. Scale-space properties of regularization methods. In M. Nielsen, P. Johansen, O. F. Olsen, and J. Weickert, editors, *Scale-Space Theories in Computer Vision*, number 1682 in Lecture Notes in Computer Science, Berlin, Germany, September 1999. Springer.

[32] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection.* Wiley, 1987.

[33] M. Rumpf and R. Strzodka. Nonlinear diffusion in graphics hardware. Research report 39, Rheinische Friedrich-Wilhelms-Universität, 2001.

[34] O. Scherzer and J. Weickert. Relations between regularization and diffusion filtering. Technical Report DIKU-98/23, Dept. of Computer Science, University of Copenhagen, Denmark, Oct 1998.
Available at `http://www.diku.dk/research/published/98-23.ps`.

[35] V. Smutný. Analysis of rainbow range finder errors. In V. Hlaváč and T. Pajdla, editors, *1st Czech Pattern Recognition Workshop*, pages 59–66, Temešvár u Písku, Nov.4–6 1993. Czech Pattern Recognition Society.

[36] V. Smutný and T. Pajdla. Rainbow Range Finder and its implementation at the CVL. Technical Report Nr. K335-96-130, Czech Technical University, Prague, May 1996.

[37] J. Sporring and J. Weickert. On generalized entropies and scale-space. In B. ter Haar Romeny, L. Florack, J. Koenderink, and M. Viergever, editors, *Scale-Space Theory in Computer Vision*, volume 1252 of *Lecture Notes In Computer Science*, pages 53–64. Springer, Berlin, 1997.

[38] J. Sporring and J. Weickert. Information measures in scale-spaces. *IEEE Transactions on Information Theory*, 45:1051–1058, 1999.

[39] J. Tajima and M. Iwakawa. 3-D data acquisition by rainbow range finder. In *Proc. 10th Int. Conf. on Pattern Recognition*, volume 1, pages 309–313, June 1990.

[40] B. M. ter Haar Romeny, editor. *Geometry-Driven Diffusion in Computer Vision.* Kluwer Academic Publishers, 1994.

[41] A. Tikhonov and A. Goncharsky, editors. *Ill-Posed Problems in the Natural Sciences.* MIR Publishers, Moscow, 1987. Translated from Russian.

[42] J. Weickert. Nonlinear diffusion scale-spaces: From the continuous to the discrete setting. In M.-O. Berger, R. Deriche, I. Herlin, J. Jaffré, and J.-M. Morel, editors, *ICAOS '96: Images, Wavelets and PDEs*, volume 219 of *Lecture Notes in Control and Information Sciences*, pages 111–118. Springer, 1996.
Available at `http://www.cvgpr.uni-mannheim.de/weickert/publications.html`.

[43] J. Weickert. Theoretical foundations of anisotropic diffusion in image processing. *Computing, Suppl. 11*, pages 221–236, 1996.

[44] J. Weickert. A review of nonlinear diffusion filtering. In B. M. ter Haar Romeny, L. Florack, J. Koenderink, and M. Viergever, editors, *Scale-Space Theory in Computer Vision*, volume 1252 of *Lecture Notes in Computer Science*, pages 3–28. Springer, 1997. Invited paper.
Available at `http://www.cvgpr.uni-mannheim.de/weickert/publications.html`.

[45] J. Weickert. *Anisotropic Diffusion in Image Processing.* European Consortium for Mathematics in Industry. B. G. Teubner, Stuttgart, 1998.

[46] J. Weickert. Coherence-enhancing diffusion filtering. *International Journal of Computer Vision*, 31(2-3):111–127, 1999.

[47] J. Weickert. Coherence-enhancing diffusion of colour images. *Image and Vision Computing*, 17:201–212, 1999.

[48] J. Weickert. Nonlinear diffusion filtering. In P. G. B. Jaehne, H. Haussecker, editor, *Handbook of Computer Vision and Applications*, volume 2: Signal Processing and Pattern Recognition, pages 423–450. Academic Press, San Diego, 1999.

[49] J. Weickert. Efficient image segmentation using partial differential equations and morphology. Technical Report 3/2000, Computer Vision, Graphics, and Pattern Recognition Group, Department of Mathematics and Computer Science, University of Mannheim, Germany, February 2000.
Available at `http://www.cvgpr.uni-mannheim.de/weickert/Papers/TR_3_00.ps.gz`.

[50] J. Weickert and B. Benhamouda. Why the Perona-Malik filter works. Technical Report DIKU-97/22, Dept. of Computer Science, University of Copenhagen, Denmark, 1997.
Available at `http://www.cvgpr.uni-mannheim.de/weickert/publications.html`.

[51] J. Weickert, S. Ishikawa, and A. Imiya. Linear scale-space has first been proposed in Japan. *Journal of Mathematical Imaging and Vision*, 10(3):237–252, 1999.

[52] J. Weickert and H. Scharr. A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance. Technical Report 4/2000, Computer Vision, Graphics, and Pattern Recognition Group, Department of Mathematics and Computer Science, University of Mannheim, Germany, February 2000.

[53] J. Weickert, B. ter Haar Romeny, and M. A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7:398–410, 1998.
Available at `http://www.cvgpr.uni-mannheim.de/weickert/publications.html`.

[54] J. Weickert, K. Zuiderveld, B. ter Haar Romeny, and W. Niessen. Parallel implementations of AOS schemes: A fast way of nonlinear diffusion filtering. In *Proc. 1997 IEEE International Conference on Image Processing*, volume 3, pages 396–399, 1997.

[55] R. Whitaker and G. Gerig. Vector-valued diffusion. In ter Haar Romeny [40], pages 93–134.

[56] A. P. Witkin. Scale space filtering. In A. Bundy, editor, *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 1019–1023, Karlsruhe, Germany, Aug 1983. William Kaufmann.

[57] Y.-L. You and M. Kaveh. Blind image restoration by anisotropic regularization. *IEEE Transactions on Image Processing*, 8(3):396 –407, Mar 1999.

Index