Czech Pattern Recognition Workshop 2000, Tomáš Svoboda (Ed.)
Peršlák, Czech Republic, February 2–4, 2000
Czech Pattern Recognition Society

Enhancing Monotonicity by Nonlinear Diffusion of Image Derivatives

Pavel Mrázek

Center for Machine Perception
Czech Technical University, Faculty of Electrical Engineering
Technická 2, 166 27 Praha 6, Czech Republic
www: http://cmp.felk.cvut.cz
e-mail: mrazekp@cmp.felk.cvut.cz

Abstract *We consider the task of filtering the noise from images and other types of inputs which are assumed to be piecewise continuous and piecewise monotone. We show that nonlinear diffusion of the data, a powerful filtering method, is too restrictive for such a case, leading to piecewise constant functions. We claim that the piecewise monotonicity can be enhanced by nonlinear diffusion of first partial derivatives of the input data. The method is developed in this paper; we introduce the algorithms and present experimental results.*

1 Introduction

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

$$\begin{array}{lll} & f(x_1,y) \leq f(x_2,y) & \forall (x_1,y),(x_2,y) \in A_k, \ \ x_1 < x_2 \\ \text{or} & f(x_1,y) \geq f(x_2,y) & \forall (x_1,y),(x_2,y) \in A_k, \ \ x_1 < x_2 \end{array} \tag{1}$$

$$\begin{array}{lll} & f(x,y_1) \leq f(x,y_2) & \forall (x,y_1),(x,y_2) \in A_k, \ \ y_1 < y_2 \\ \text{or} & f(x,y_1) \geq f(x,y_2) & \forall (x,y_1),(x,y_2) \in A_k, \ \ y_1 < y_2 \end{array} \tag{2}$$

The function f is sampled and represented by a 2D array of values $f_{i,j}$:

$$f_{i,j} = h_{i,j} * f + n, \ \ x_i = i \cdot \Delta x, \ \ y_j = j \cdot \Delta y \tag{3}$$

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.

In the discrete case the piecewise monotonicity assumption can be restated as follows: if K is the smallest number of connected sets A_k needed to partition the function domain so that the discrete function f is continuous[1] and monotone on each A_k, we require that K is much smaller then 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 f so as to enforce the piecewise monotonicity (reduce the number K described above) while preserving important discontinuities or edges.

To give a real world example of the situation where the piecewise monotonicity assumption could be appropriate, we may mention the range data for 3D reconstruction in computer vision. The function 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 $f_{i,j}$.

In this paper we concentrate on the possibility to enhance piecewise monotonicity of the data by nonlinear diffusion. We first review the basics of nonlinear diffusion, a powerful filtering method, in section 2, and present its simplest discrete algorithm in section 3. Section 4 discusses why classical NL diffusion is not suited for monotonicity enhancement, and suggests to remedy the problems by moving to NL diffusion of first directional derivatives of the input data. This method is developed in sections 5–7, some experiments follow in section 8.

The author gratefully acknowledges the support of grant 201/97/0437 of the Grant Agency of the Czech Republic, grant no. 30/99071/333 of the Czech Technical University, and grant Aktion Österreich - Tschechien 23p16.

[1] 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.

2 Nonlinear diffusion

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, see [5]) which not only removes noise but also blurs and dislocates edges. A good introduction to NL diffusion can be found in [10, 11], Weickert in [12] gives a rich survey of the literature. The following brief presentation is adapted loosely from Weickert *et al.* [13].

We take the filter of Catté *et al.* [3], a regularization of the pioneering Perona–Malik model [7], as a typical representative of a well founded nonlinear diffusion process. With this scheme, the filtered image is found as a solution to the equation

$$\partial_t u = \nabla \cdot \left(g(|\nabla u_\sigma|^2) \nabla u \right) \tag{4}$$

with the original image f as the initial state, and the reflecting boundary conditions,

$$u(\mathbf{x}, 0) = f(\mathbf{x}), \quad (\partial_t u) \cdot n = 0 \text{ on } \partial\Omega \tag{5}$$

where n denotes the normal to the image boundary $\partial\Omega$. In words, the equation (4) expresses the fact that the value of $u(\mathbf{x}, t)$ changes with time according to the flow to and from the neighbourhood of $\mathbf{x}$; this flow depends on the image gradient ∇u and its amount is controlled by the function g of the smoothed gradient ∇u_σ (smoothing makes the filter insensitive to noise at scales smaller then σ); no flow passes through the image boundary.

The diffusivity function $g(s)$ is typically constructed to be positive everywhere but rapidly and monotonically decreasing for $s > 0$; to give an example, the diffusivity

$$g(s) = \begin{cases} 1 & \text{for } s < 0 \\ 1 - \exp\left(\frac{-3.315}{(s/\lambda)^4} \right) & \text{for } s \geq 0 \end{cases} \tag{6}$$

was used in [13]. The parameter λ in this formula can be understood as a threshold of function continuity: if $s \ll \lambda$, $g(s)$ is almost one and the position of such a small gradient $s = |\nabla u_\sigma|$ is considered to belong to a continuous region where diffusion (or smoothing) is encouraged. On the other hand, g is close to zero for $s \gg \lambda$, almost no diffusion takes place at positions of a larger gradient. This way the small-scale noise in otherwise homogeneous regions is removed, whereas important discontinuities between regions remain stable over long periods of the diffusion process.

The positivity of g guarantees that the solution $u(., t)$ converges to a constant for $t \to \infty$. To obtain nontrivial results, a finite stopping time T has to be set.

3 Discrete nonlinear diffusion

To be suitable for numerical computations with sampled data, the continuous equation (4) has to be discretized. Let us start in one dimension only, where the equation (4) reads

$$\partial_t u = \partial_x \left(g(|\partial_x u_\sigma|^2) \partial_x u \right). \tag{7}$$

For discrete data u_i^k (approximating 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 (7) becomes

$$\frac{u_i^{k+1} - u_i^k}{\tau} = \sum_{j \in \mathcal{N}(i)} \frac{g_{ij}^k}{\Delta x^2} (u_i^k - u_j^k) \tag{8}$$

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 (8) is called the explicit discretization scheme of (7). It can be summarized into the following iterative formula:

$$u^{k+1} = \left(I + \tau A(u^k) \right) u^k, \tag{9}$$

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

$$a_{i,j} = \begin{cases} \frac{g_{ij}^k}{\Delta x^2} & \text{for } j \in \mathcal{N}(i) \\ -\sum_{n \in \mathcal{N}(i)} \frac{g_{in}^k}{\Delta x^2} & \text{for } j = i \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

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

$$u^{k+1} = \left(I + \tau A_x(u^k) + \tau A_y(u^k) \right) u^k \tag{11}$$

and $A_x(u^k)$ and $A_y(u^k)$ are matrices containing information about the diffusivities between individual pixels in the directions of axes x and y, respectively[2].

The explicit (or Euler) discretization scheme used in this section is the most straightforward but requires a small time step τ (and thus more iterations) in order to be stable; more efficient and more complicated, absolutely stable procedures like the semi-implicit scheme and the additive operator splitting have been introduced in [13].

4 Monotonicity enhancing NL diffusion

Being smoothed more inside homogeneous regions, the function u solving the equation (4) tends to piecewise constant as the time t increases. This is illustrated on a simple function in figure 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 of the piecewise-constant model for noise removal using minimization of the total variation of the image[3] were observed by

[2] The pixels of u must be arranged into a single column vector to allow the matrix multiplication.

[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. [8, 9].

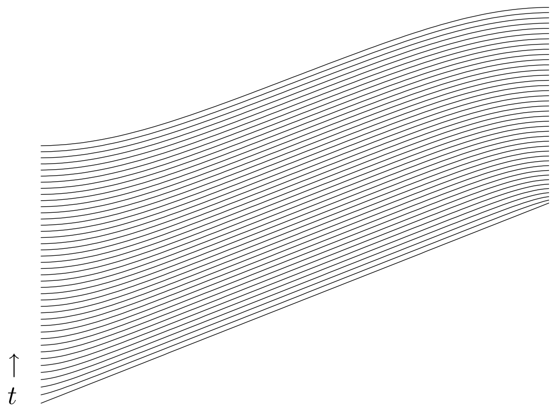

Figure 1: Classical nonlinear diffusion approaches a (piecewise) constant function; this phenomenon can be observed first near the ends of growing function segments.

Chambolle and Lions in [4]. The authors there suggest to alleviate the problem by introducing second order terms (like the total variation of the image gradient) into the functional to be minimized.

In this paper we try to exploit the following simple idea: if we differentiate the image first and run the diffusion on the arrays of partial derivatives instead of the grey values, the piecewise smoothing of derivatives leads (after integration) to an image piecewise monotone, for higher t approaching a function piecewise linear. This behaviour is demonstrated in figure 4 bottom right: where nonlinear diffusion simplifies the image into segments of similar grey levels, the procedure using derivatives (the main topic of this paper) creates patches of similar *trends*, which can successfully approximate a large class of images and other types of inputs and will be preferred if the piecewise monotonicity of the data is assumed and desired.

5 From data to partial derivatives

Consider the two-dimensional situation: a function $f(x, y)$ is sampled and represented by values $f_{i,j}$ at positions (x_i, y_j), $x_i = i \cdot \Delta x$, $y_j = j \cdot \Delta y$, $i = 1, \dots, N_i$, $j = 1, \dots, N_j$; 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 v and w:

$$\frac{\partial f(x,y)}{\partial x}\Big|_{x_i,y_j} \approx v_{i,j} \equiv \frac{f_{i+1,j} - f_{i,j}}{\Delta x}, \qquad i = 1, \dots, N_i - 1,\ j = 1, \dots, N_j \tag{12}$$

$$\frac{\partial f(x,y)}{\partial y}\Big|_{x_i,y_j} \approx w_{i,j} \equiv \frac{f_{i,j+1} - f_{i,j}}{\Delta y}, \qquad i = 1, \dots, N_i,\ j = 1, \dots, N_j - 1. \tag{13}$$

The arrays v, 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,

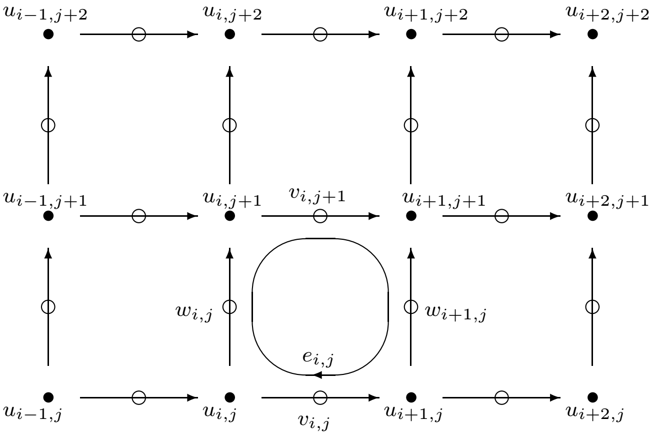

Figure 2: Geometry of the discrete image: the partial derivatives of the data u in the direction of axes x, y are approximated by finite differences forming arrays v and w, respectively.

$$\oint_{\mathcal{C}} \nabla f(\mathbf{r}) \cdot \mathrm{d}\mathbf{r} = 0. \tag{14}$$

In the discrete image any closed curve is composed of the elementary closed curves passing through four pixels[4] as illustrated in figure 2, and the observation (14) transforms into

$$\forall i, j: \quad e_{i,j} \equiv w_{i,j} + v_{i,j+1} - w_{i+1,j} - v_{i,j} = 0. \tag{15}$$

While this constraint is satisfied automatically by (12)–(13), we have to be more careful about it during the diffusion process.

6 The diffusion algorithm

The continuous equations for nonlinear diffusion have been discretized in section 3. Simply rewriting (11) using v, w instead of u, we obtain the following set of equations:

$$v^{k+1} = \big(I + \tau A_x(v^k, w^k) + \tau A_y(v^k, w^k)\big)\, v^k \tag{16}$$

$$w^{k+1} = \big(I + \tau B_x(v^k, w^k) + \tau B_y(v^k, w^k)\big)\, w^k \tag{17}$$

Here v, w are column vectors of generally different size, $(N_i - 1) \cdot N_j$, and $N_i \cdot (N_j - 1)$, respectively; the matrices A, B 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 2), we find it reasonable to assign these two connections the same diffusivity, namely $g_{i,j}$. This is why the matrices A and B depend on both v^k and w^k; the equations (16), (17) will be coupled through the common array of diffusivities and many elements of A_x and B_y will be identical (similarly for the other pair of directions and A_y, B_x, only the common point of the connections $v_{i,j}$ — $v_{i,j+1}$ and $w_{i,j}$ — $w_{i+1,j}$ does not coincide with any input datum u, and we will denote the diffusivity of that position $g_{i+\frac{1}{2},j+\frac{1}{2}}$).

There are many possibilities how to assemble the information from the smoothed, regularized versions $\tilde{v}, \tilde{w}$ of the

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

arrays v, w into the common diffusivities $g_{i,j}$. Proceeding most directly from (4) to the diffusion of derivatives, we obtain the following:

$$\begin{aligned} g\left(|\nabla^2 u_\sigma|^2\right) &= g\left(|\nabla\cdot(\nabla u_\sigma)|^2\right) \qquad (18)\\ &\approx g\left(|\nabla\cdot(\tilde v,\tilde w)|^2\right) = g\left(\left|\frac{\tilde v}{\partial x}+\frac{\tilde w}{\partial y}\right|^2\right)\approx\\ g_{i,j} &\equiv g\left(|\tilde v_{i,j}-\tilde v_{i-1,j}+\tilde w_{i,j}-\tilde w_{i,j-1}|^2\right)\end{aligned}$$

In the experiments presented below, the equation (6) is used to define the function $g(\cdot)$.

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 (18) 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

$$\begin{aligned} g\left(|\nabla u_\sigma|^2\right) &\approx \qquad (19)\\ g_{i,j} &\equiv g\left(\left|\frac{\tilde v_{i-1,j}+\tilde v_{i,j}}{2}\right|^2+\left|\frac{\tilde w_{i,j-1}+\tilde w_{i,j}}{2}\right|^2\right)\\ g_{i+\frac12,j+\frac12} &\equiv g\left(\left|\frac{\tilde v_{i,j}+\tilde v_{i,j+1}}{2}\right|^2+\left|\frac{\tilde w_{i,j}+\tilde w_{i+1,j}}{2}\right|^2\right)\end{aligned}$$

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 (16)–(17) do not guarantee that the constraint (15) is satisfied. In the remaining part of this section we try to enforce the necessary properties of the arrays of partial derivatives.

Denote by $\hat z = [v^k, w^k]^T$ the result of the diffusion process at time t_k. We seek a solution z as close as possible to $\hat z$ while obeying the constraint (15) which can be written in matrix form as

$$C\,z = 0 \qquad (20)$$

where C is a $[(N_i-1)\cdot(N_j-1)]\times[(N_i-1)\cdot N_j+N_i\cdot(N_j-1)]$ sparse matrix with four nonzero entries in each row. The rigorous way to solve this problem would be to find z as a projection of $\hat z$ into the null space of matrix C; such solution would minimize the norm $\|z-\hat z\|_2$, see e.g. [6]. However, this mathematically correct solution involves the construction of an orthonormal basis of the null space of C, and full matrix multiplication (processes of complexities $O(N^3)$ and $O(N^2)$, respectively, where N denotes the larger of the dimensions of C). Already for small images, these matrix computations become infeasible.

As a viable alternative, we choose to restore the property (15) by the following iterative algorithm:

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]:
$$\begin{aligned} v_{i,j} &= v_{i,j}+(e_{i,j}-e_{i,j-1})/c\\ w_{i,j} &= w_{i,j}-(e_{i,j}-e_{i-1,j})/c\end{aligned}$$
with the obvious modifications at the image boundary.
3. If $\max|e_{i,j}|$ is smaller than a given threshold, finish; otherwise go to 1.

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 usually to a faster convergence. The complexity of one iteration of this algorithm is only $O(N)$ with $N \approx 2N_iN_j$ the number of elements of the arrays v, w. Although several iterations are necessary (with the Lena image below, about 12 iterations were needed to satisfy the error threshold of 1 per cent of the function range), this still compares favorably with the complexity of the matrix method.

7 From derivatives back to data

Let us now assume that the filtered arrays v, w are available, and that they contain correct values in the sense of condition (15). These arrays contain (redundantly) all the information needed for the reconstruction of the image u up to a scalar u_0 added to function values. The integration can be performed by the following algorithm:

1. Reconstruct the first row:
```
$\hat u_{1,1} = 0$
for $i = 2,\dots,N_i$
        $\hat u_{i,1} = \hat u_{i-1,1} + v_{i-1,1}$
```
2. Reconstruct the columns:
```
for $i = 1,\dots,N_i$
    for $j = 2,\dots,N_j$
        $\hat u_{i,j} = \hat u_{i,j-1} + w_{i,j-1}$
```
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 value of the original, f, and the filtered image, u, remain equal:

$$u_0 = \frac{1}{N_i\cdot N_j}\left[\sum_{i=1}^{N_i}\sum_{j=1}^{N_j} f_{i,j} - \sum_{i=1}^{N_i}\sum_{j=1}^{N_j}\hat u_{i,j}\right] \qquad (21)$$

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

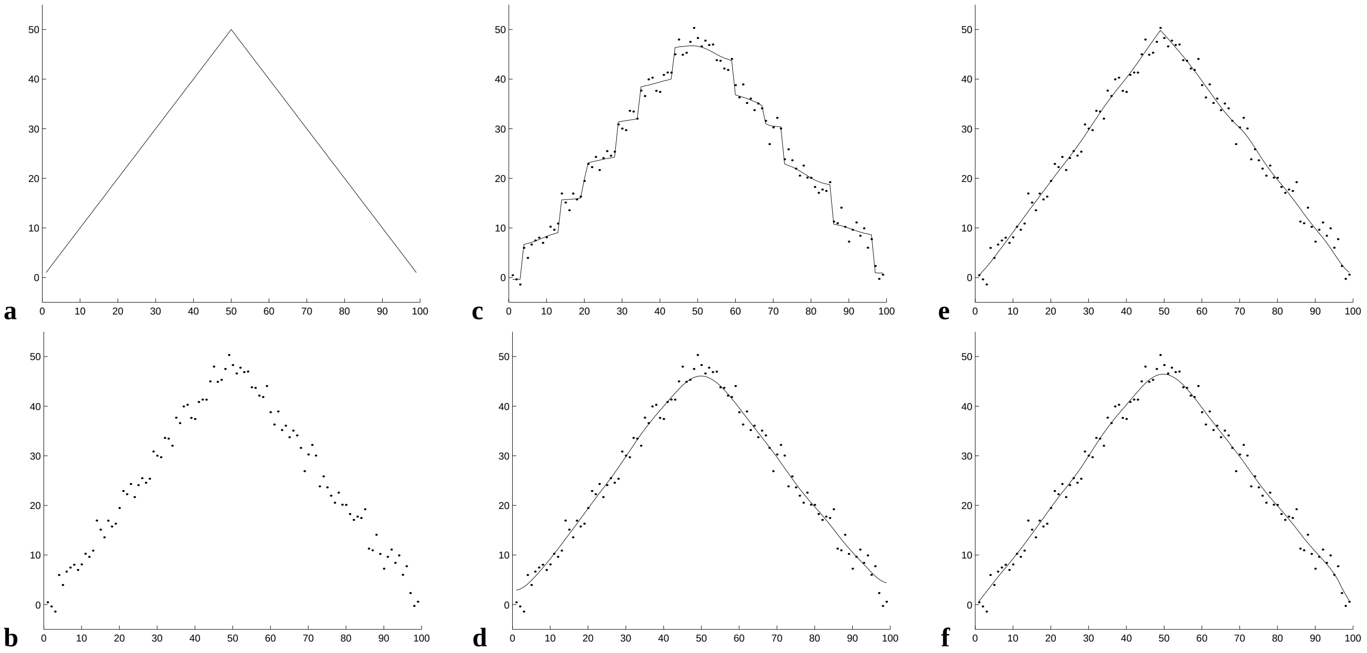

Figure 3: Gaussian noise was added to a triangle function (a) to obtain the noisy data (b), used to test the diffusion filtering methods. The filtering results (continuous lines) are shown together with the noisy input (dots). (c) Ordinary nonlinear diffusion, $\sigma = 1, \lambda = 1, T = 10$. (d) Ordinary nonlinear diffusion, $\sigma = 1, \lambda = 3, T = 10$. (e) Filtering by nonlinear diffusion of first derivatives, $\sigma = 3, \lambda = 0.05, T = 10$. The diffusion is controlled by the second order derivatives as described by (18). (f) Filtering by nonlinear diffusion of first derivatives, $\sigma = 1, \lambda = 1, T = 10$. The diffusion controlled by first derivatives according to (19).

8 Experiments

The results of all the methods of nonlinear diffusion mentioned in this paper applied to a simple 1D function are shown for comparison in figure 3. In the center column 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. On the right, two results of the nonlinear diffusion of first derivatives are shown for comparison: the lower 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 upper one, where the diffusion was dominated by a function of second derivatives, is able (for carefully chosen parameters) to precisely locate the corner in the function values, i.e. the discontinuity of the second derivative.

Figure 4 left 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 the piecewise monotonicity can be assumed. The results of the classical nonlinear diffusion from equation (4) and the derivative-based diffusion are shown for comparison in the center column of figure 4; both methods perform well in removing the noise, the diffusion using first derivatives is better at preserving the gradual transitions of light intensities. On the right, 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.

9 Conclusion

We have presented a method for piecewise monotone approximation which allows to remove noise and enhance the trends while preserving important discontinuities present in the input data. The task is accomplished by nonlinear vector-valued diffusion of partial derivatives of image data. 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.

In the future we would like to analyze the method in more detail, and develop a general mechanism which would allow to adapt the parameters of the procedure autonomously to obtain the best results for any given input data.

Acknowledgement

The author would like to express his gratefulness to Mirko Navara for his constant support, assistance, and valuable comments on the manuscript.

References

[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.

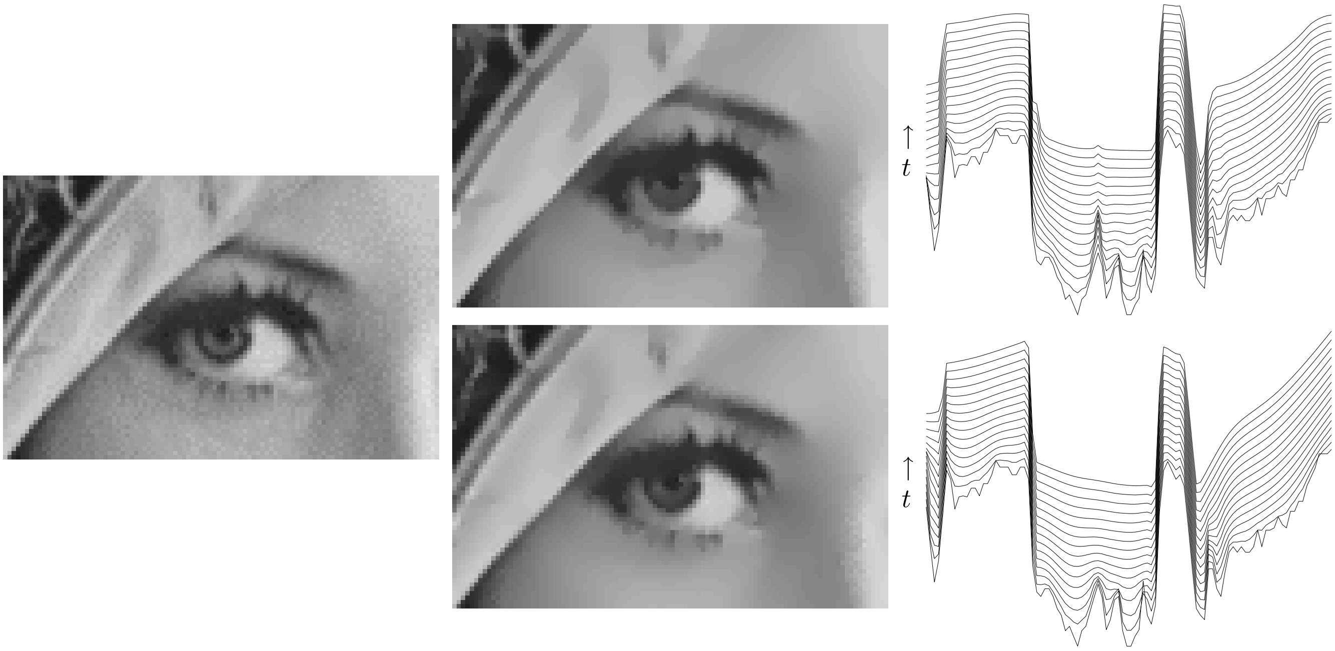

Figure 4: Left: Input image, a detail of Lena.
Center top: ordinary nonlinear diffusion of the input image; center bottom: the same using nonlinear diffusion of first partial derivatives. Both methods were run with the parameters $\sigma = 1,\ \lambda = 1,\ \tau = 1,\ T = 5$.
Right top and bottom: one 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 at the top approach a (piecewise) constant function; the nonlinear diffusion of first partial derivatives (at the bottom) leads to a function piecewise linear.
These images can be also seen at http://cmp.felk.cvut.cz/˜mrazekp/Diffusion/diffusion.html.

[3] 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.

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

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

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

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

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

[13] 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.