Restoration of Images with Rotated Shapes

S. Setzer and G. Steidl and T. Teuber [*]

*Dedicated to Professor Manfred Tasche
in occasion of his 65th birthday*

May 19, 2008

Abstract

Methods for image restoration which respect edges and other important features are of fundamental importance in digital image processing. In this paper, we present a novel technique for the restoration of images containing rotated (linearly transformed) rectangular shapes which avoids the round-off effects at vertices produced by known edge-preserving denoising techniques. Following an idea of Berkels et al. our approach is also based on two steps: the determination of the angles related to the rotated shapes and a subsequent restoration step which incorporates the knowledge of the angles. However, in contrast to Berkels et al., we find the smoothed rotation angles of the shapes by minimizing a simple quadratic functional without constraints which involves only first order derivatives so that we finally have to solve only a linear system of equations. Moreover, we propose to perform the restoration step either by quadratic programming or by solving an anisotropic diffusion equation. We focus on a discrete approach which approximates derivatives by finite differences. Particular attention is paid to the choice of the difference filters. We prove some relations concerning the preservation of rectangular shapes for our discrete setting. Finally, we present numerical examples for the denoising of artificial images with rotated rectangles and parallelograms and for the denoising of a real-world image.

1 Introduction

In image denoising one is typically interested in removing noise while preserving important structures such as edges. Since this goal cannot be achieved with linear filters various nonlinear strategies have been proposed in recent years, e.g., wavelet-based methods [8], stochastic methods [19], variational methods like the Rudin-Osher-Fatemi model [11], PDE-based methods like

[*]University of Mannheim, Dept. of Mathematics and Computer Science

1

the isotropic nonlinear diffusion of Perona and Malik [9] [1], or Weickert's anisotropic edge enhancing diffusion [16]. In this paper, we focus on variational and PDE-based approaches. Most of these methods cause a significant rounding artefact at corners formed by sharp edges. A typical example is shown in Fig. 1.
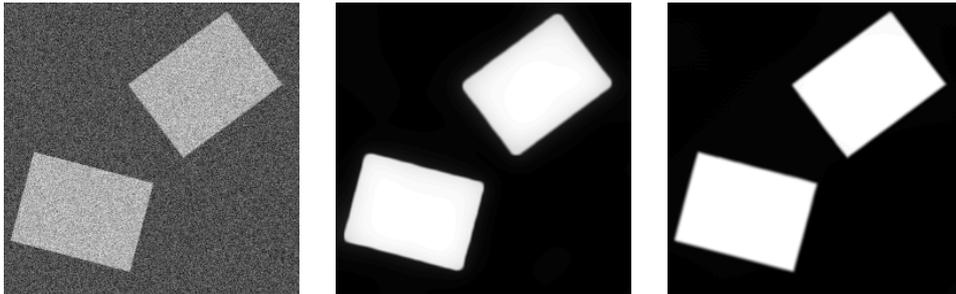
Figure 1: Left: noisy image currupted by white Gaussian noise with standard deviation 50. Middle: denoised image using edge enhancing diffusion, see [16]. Right: denoised image using the method proposed in this paper.

One suitable approach to overcome this drawback was the anisotropic modification of the Rudin-Osher-Fatemi model pointed out by Esedoglu and Osher [6], see also [7, 5], and further developed by Berkels et al. [3]. Note that a related approach was given from the PDE point of view by Weickert [15].
In [3], the authors deal with images f containing noisy rotated rectangular shapes. They suggest to find the denoised version u of f by computing the minimizer of the functional

$$F(u, \alpha) \quad := \quad \frac{1}{2} \int_{\Omega} (f - u)^2 \, dxdy + \lambda \int_{\Omega} \|R(\alpha)\nabla u\|_1 \, dxdy$$

$$+ \frac{1}{2} \int_{\Omega} \mu_1 |\nabla \alpha|^2 + \mu_2 |\triangle \alpha|^2 \, dxdy$$

with the rotation matrices

$$R(\alpha(x, y)) := \begin{pmatrix} \cos(\alpha(x, y)) & \sin(\alpha(x, y)) \\ -\sin(\alpha(x, y)) & \cos(\alpha(x, y)) \end{pmatrix}. \tag{1}$$

More precisely, they propose to find u by iterating the following alternating procedure: for fixed $u(x, y)$ compute the minimizer of

$$F_u(\alpha) := \int_{\Omega} \|R(\alpha)\nabla u\|_1 \, dxdy + \frac{1}{2} \int_{\Omega} \mu_1 |\nabla \alpha|^2 + \mu_2 |\triangle \alpha|^2 \, dxdy, \tag{2}$$

---

[1]Sometimes these processes are already denoted as *anisotropic*. In our nomenclature, this notion is reserved for processes driven by matrix-valued diffusion tensors.

and for fixed $\alpha(x, y)$ find the minimizer of

$$E_\alpha(u) := \frac{1}{2} \int_\Omega (f - u)^2 \, dxdy + \lambda \int_\Omega \|R(\alpha)\nabla u\|_1 \, dxdy. \qquad (3)$$

The main idea here is that when rotating the gradient the smallest L_1-norm is attained for the rotated gradient being parallel to one of the axes. Consequently, the solution of $F_u(\alpha)$ assigns to each point a smoothed angle α, representing the orientation of the gradients in the image. Using this information in the second term of (3), we see that minimizing $E_\alpha(u)$ leads to good denoising results for objects which are aligned in the directions α and $\alpha + \frac{\pi}{2}$. The functional $F_u(\alpha)$ depends on α and also on the trigonometric functions $\cos \alpha$ and $\sin \alpha$ of α. Moreover, we see that $\|R(\alpha(x, y))\nabla u(x, y)\|_1$ has the same value for $\alpha(x, y) + k\frac{\pi}{2}$, $k \in \mathbb{Z}$ so that $E_\alpha(u)$ depends only on α modulo $\frac{\pi}{2}$. Corresponding restrictions on α while computing a minimum of F_u in (2) were not addressed in [3]. To cope with the significant loss of contrast in this process, the authors propose to apply in addition Bregman iterations, a kind of inverse scale space method.

The aim of this paper is to present a simple model for the restoration of images containing rotated (linearly transformed) rectangles. The approach is based on the idea of Berkels et al. in so far as we also iterate two steps: an angle adaptation step and a restoration step which takes the former angle computation into account. In the angle adaptation step we find the smoothed rotation angles of the shapes contained in the image. To this end, we minimize a simple quadratic functional without constraints so that we finally have to solve only a linear system of equations. Attention is paid to the fact that the angles are only needed modulo $\frac{\pi}{2}$. For the restoration step we propose two methods, namely either to minimize a discrete version of (3) by quadratic programming or to solve an anisotropic diffusion equation related to the Euler-Lagrange equation of (3) by finite difference methods. In both steps, we focus on discrete techniques which approximate derivatives by forward differences. Therefore, we have to pay attention to the careful choice of the difference filters. Moreover, we consider some results of Esedoglu and Osher [6] from our discrete point of view. On the one hand, this makes the approaches simpler and also manageable for other operators than the gradient. On the other hand structures like Wulff shapes in $\mathbb{R}^m$ do not carry over to $\mathbb{Z}^m$.

This paper is organized as follows: Section 2 describes our mathematical model. In particular, we explain the angle adaptation in Subsection 2.1 and the two restoration variants in the Subsections 2.2 and 2.3. In Section 3, we prove some useful relations concerning the preservation of discrete rectangular shapes. Finally, in Section 4, we present various numerical examples for the denoising of artificial images with rotated rectangles and parallelograms as well as for the denoising of a real-world image. The paper concludes with

a summary and a sketch of ongoing and future work in Section 5.

2 Mathematical model

Our denoising algorithm is based on iterations of two steps, the angle adaptation step and the restoration step which we describe in the following subsections. For the restoration step, we propose either to apply quadratic programming methods or to solve an anisotropic diffusion equation.

In our discrete setting, we assume for convenience that $\mathbf{f} : \{0, \dots, n-1\} \times \{0, \dots, n-1\} \to \mathbb{R}$ is a quadratic image. Alternatively, we may reshape $\mathbf{f} = (f(x,y))_{x,y=0}^{n-1}$ columnwise into a vector $f : \{0, \dots, N-1\} \to \mathbb{R}$, where $N = n^2$.

2.1 Angle adaptation

Instead of the functional (2) one may try to compute the smoothed rotation angles α by minimizing the simpler quadratic functional

$$\int\limits_{\Omega} (\alpha_u - \alpha)^2 \, dxdy + \mu \int\limits_{\Omega} |\nabla \alpha|^2 \, dxdy, \tag{4}$$

where $\alpha_u(x,y)$ is the angle of the gradient of u at (x,y). Note that

$$|\nabla \alpha(x,y)|^2 = |\nabla \cos(\alpha(x,y))|^2 + |\nabla \sin(\alpha(x,y))|^2. \tag{5}$$

Unfortunately, the functional (4) does not take α modulo $\frac{\pi}{2}$ into account so that we have to look for corresponding modifications in the rest of this subsection.

To find the desired rotation angles α we first have to compute the rotation angles α_u of the given image $u \in \mathbb{R}^{n,n}$ and then to smooth them. We focus on the discrete setting.

Computation of α_u. We compute $\alpha_u(x,y)$ as the angle of the discrete gradient of $\mathbf{u}$ in (x,y), $(x,y) \in \{0, \dots, n-1\} \times \{0, \dots, n-1\}$. More precisely, we compute this angle with respect to the smoothed image $\mathbf{u}_\sigma := K_\sigma * \mathbf{u}$ obtained by convolving $\mathbf{u}$ with the Gaussian K_σ of standard deviation σ. As discrete partial derivatives of $\mathbf{u}_\sigma$ we use

$$\mathbf{u}_{\sigma,x} := H_1 \, \mathbf{u}_\sigma \, H_0^{\mathrm{T}}, \quad \mathbf{u}_{\sigma,y} := H_0 \, \mathbf{u}_\sigma \, H_1^{\mathrm{T}} \tag{6}$$

with

$$H_0 := \frac{1}{2} \begin{pmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & \ddots & & \\ & & & 1 & 1 \\ & & & & 2 \end{pmatrix}, \quad H_1 := \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & & \\ & & & -1 & 1 \\ & & & & 0 \end{pmatrix}.$$

In other words, $\mathbf{u}_{\sigma,x}$ is computed by taking forward differences of $\mathbf{u}_\sigma$ in vertical direction, the usual smoothing in horizontal direction, and assuming Neumann boundary conditions. If we suppose pixel distances h_x, h_y not equal to 1, then H_1 must be multiplied by $1/h_x$ and H_1^{T} by $1/h_y$ to obtain consistent discretizations of the derivatives in x- and y-directions in (6). Note that H_0, H_1 are also the filter matrices related to the undecimated Haar wavelet transform, cf. [8]. Alternatively, one can also apply symmetric differences which leads to similar numerical results. In the vector reshaped version, (6) reads

$$\nabla u_\sigma = \begin{pmatrix} D_x \\ D_y \end{pmatrix} u_\sigma = \begin{pmatrix} H_0 \otimes H_1 \\ H_1 \otimes H_0 \end{pmatrix} u_\sigma, \tag{7}$$

where $\otimes$ denotes the tensor product of matrices. For $(x,y) \in \{0,\ldots,n-1\} \times \{0,\ldots,n-1\}$, let $|\nabla u_\sigma(x,y)| := (u_{\sigma,x}^2(x,y) + u_{\sigma,y}^2(x,y))^{1/2}$. If $|\nabla u_\sigma(x,y)| = 0$ we set $\alpha_u(x,y) := 0$, and otherwise

$$\alpha_u(x,y) = \begin{cases} \arccos \frac{u_{\sigma,x}(x,y)}{|\nabla u_\sigma(x,y)|} & \text{if } u_{\sigma,y}(x,y) \geq 0, \\ 2\pi - \arccos \frac{u_{\sigma,x}(x,y)}{|\nabla u_\sigma(x,y)|} & \text{if } u_{\sigma,y}(x,y) < 0. \end{cases}$$

Next, we want to smooth α_u in an appropriate way.

Smoothing of α_u. Since we want to take only α modulo $\frac{\pi}{2}$ into account, it makes sense to use the $\frac{\pi}{2}$-periodic functions

$$c_u(x,y) := \cos(4\alpha_u(x,y)), \qquad s_u(x,y) = \sin(4\alpha_u(x,y))$$

in the smoothing process. Now we are looking for vectors $(c_v, s_v)^{\mathrm{T}}$ such that

$$|\nabla u| \begin{pmatrix} c_u \\ s_u \end{pmatrix} \approx |\nabla u| \begin{pmatrix} c_v \\ s_v \end{pmatrix}.$$

We do not require that $c_v^2 + s_v^2 = 1$. Having (5) in mind, we solve

$$\arg\min_{c_v, s_v} \sum_{(x,y) \in \mathbb{Z}_n^2} |\nabla u|^2 \left| \begin{pmatrix} c_u \\ s_u \end{pmatrix} - \begin{pmatrix} c_v \\ s_v \end{pmatrix} \right|^2 + \mu(|\nabla c_v|^2 + |\nabla s_v|^2).$$

Let I_N denote the $N \times N$ identity matrix. Reshaping the matrices columnwise and using $G := \operatorname{diag}(|\nabla u|)$ with the discrete partial derivative operators

$$\begin{pmatrix} D_x \\ D_y \end{pmatrix} := \begin{pmatrix} I_N \otimes H_1 \\ H_1 \otimes I_N \end{pmatrix}, \tag{8}$$

this is equivalent to finding the minimizers of

$$\|G(c_u - c_v)\|_2^2 + \mu \left\| \begin{pmatrix} D_x \\ D_y \end{pmatrix} c_v \right\|_2^2 \quad \text{and} \quad \|G(s_u - s_v)\|_2^2 + \mu \left\| \begin{pmatrix} D_x \\ D_y \end{pmatrix} s_v \right\|_2^2.$$

These minimizers are given by the solutions of

$$
\begin{aligned}
\left(G^{\mathrm{T}}G + \mu(D_x^{\mathrm{T}}D_x + D_y^{\mathrm{T}}D_y)\right) c_v &= G^{\mathrm{T}}G c_u, \qquad (9)\\
\left(G^{\mathrm{T}}G + \mu(D_x^{\mathrm{T}}D_x + D_y^{\mathrm{T}}D_y)\right) s_v &= G^{\mathrm{T}}G s_u.
\end{aligned}
$$

The matrix $D_x^{\mathrm{T}}D_x + D_y^{\mathrm{T}}D_y$ is the discretization of the negative Laplacian $-\Delta$ with centered differences and Neumann boundary conditions. It is a diagonal dominant matrix in $\mathbb{R}^{N,N}$ of the rank $N-1$. The matrix $G^{\mathrm{T}}G \neq 0$ is a diagonal matrix with the nonnegative diagonal entries $|\nabla u(x,y)|^2$. Then $G^{\mathrm{T}}G + \mu(D_x^{\mathrm{T}}D_x + D_y^{\mathrm{T}}D_y)$ is irreducible diagonal dominant and consequently invertible. Moreover, it is easy to check that the maximum principle

$$
|c_v(x,y)| \leq \max_{(x,y)\in\mathbb{Z}_n^2} |c_u(x,y)| \leq 1
$$

is fulfilled (similarly for s_v) by considering for example the Jacobi iteration process to solve (9). Note that we have in general $c_v^2(x,y) + s_v^2(x,y) \neq 1$.

Once we have found c_v and s_v, we compute the smoothed version α of α_u as $\alpha(x,y) := 0$ if $|\,(c_v(x,y), s_v(x,y))\,| = 0$ and

$$
\alpha(x,y) = \frac{1}{4} \begin{cases} \arccos \frac{c_v(x,y)}{|(c_v(x,y),s_v(x,y))|} & \text{if } s_v(x,y) \geq 0, \\ 2\pi - \arccos \frac{c_v(x,y)}{|(c_v(x,y),s_v(x,y))|} & \text{if } s_v(x,y) < 0. \end{cases}
$$

Now we turn to the restoration step.

2.2 Restoration by quadratic programming

We consider a discrete version of (3) which reads for columnwise reshaped images as follows:

$$
\arg\min_u \left\{ \frac{1}{2}\|f - u\|_2^2 + \lambda \| R(\alpha) \begin{pmatrix} D_x \\ D_y \end{pmatrix} u\|_1 \right\}, \qquad (10)
$$

where

$$
R(\alpha) := \begin{pmatrix} \operatorname{diag}(\cos\alpha) & \operatorname{diag}(\sin\alpha) \\ -\operatorname{diag}(\sin\alpha) & \operatorname{diag}(\cos\alpha) \end{pmatrix}
$$

and D_x and D_y are chosen as in (7). It is well-known, cf. [4, 14] and Section 3, that the minimizer $\hat{u}$ of (10) is given by $\hat{u} = f - \lambda\,(D_x^{\mathrm{T}}\,D_y^{\mathrm{T}})\,R(\alpha)^{\mathrm{T}}\,\hat{V}$, where $\hat{V}$ is the solution of the dual problem

$$
\|f - \lambda\,(D_x^{\mathrm{T}}\,D_y^{\mathrm{T}})\,R(\alpha)^{\mathrm{T}}\,V\|_2^2 \;\rightarrow\; \min \quad \text{subject to } \|V\|_\infty \leq 1. \qquad (11)
$$

This is a quadratic problem with linear constraints which can be solved by corresponding optimization methods.

6

For heavy noise, our choice of D_x and D_y due to their kernels may lead to checkerboard effects. These effects can be avoided by solving, for a small constant ν, the slightly modified problem

$$\arg\min_u \left\{ \frac{1}{2}\|f - u\|_2^2 + \lambda\|R(\alpha)\left(\begin{array}{c} D_x \\ D_y \end{array}\right) u\|_1 + \nu\|(H_1 \otimes H_1)u\|_1 \right\}.$$

By [6, 3] and the following Section 3, the restoration via minimization of (10) is useful for images containing rotated rectangles. Of course, the approach can be simply modified for rectangles which are linearly transformed by a constant transform matrix $A \in \mathbb{R}^{2,2}$:

$$\arg\min_u \left\{ \frac{1}{2}\|f - u\|^2 + \lambda\|R(\alpha)\left(A \otimes I_N\right)\left(\begin{array}{c} D_x \\ D_y \end{array}\right) u\|_1 \right\}. \qquad (12)$$

In particular, we will consider images with rotated parallelograms which can be obtained by applying a shear matrix $A(s) = \left(\begin{array}{cc} 1 & s \\ 0 & 1 \end{array}\right)$.

2.3   Restoration by anisotropic diffusion

Another way to perform the restoration step is based on the Euler–Lagrange equation of (3). For this, we replace $\varphi(x,y) := \|(x,y)\|_1 = |x| + |y|$ in (3) by the differentiable function $\varphi_\varepsilon(x,y) := \sqrt{x^2 + \varepsilon^2} + \sqrt{y^2 + \varepsilon^2}$ with $|\varepsilon| \ll 1$ and obtain

$$\widetilde{E}_\alpha(u) = \frac{1}{2}\int_\Omega (f - u)^2 \, dxdy + \lambda\int_\Omega \varphi_\varepsilon\left(R(\alpha)\nabla u\right) \, dxdy. \qquad (13)$$

The minimizer of $\widetilde{E}_\alpha(u)$ has to fulfill the Euler–Lagrange equation given by

$$0 = u - f - \lambda\operatorname{div}\left(D_\alpha(\nabla u)\nabla u\right) \qquad (14)$$

with the diffusion tensor

$$D_\alpha(\nabla u) := R(\alpha)^{\mathrm{T}}\left(\begin{array}{cc} \psi(\langle e_\alpha, \nabla u\rangle) & 0 \\ 0 & \psi(\langle e_\alpha^\perp, \nabla u\rangle) \end{array}\right) R(\alpha), \qquad (15)$$

where $\psi(x) := 1/\sqrt{x^2 + \varepsilon^2}$ and $e_\alpha := (\cos\alpha, \sin\alpha)^{\mathrm{T}}$. Then the minimizer of (14) can be obtained by computing the steady state $u(x,t)$ for $t \to \infty$ of the reaction diffusion equation

$$\partial_t u = f - u + \lambda\operatorname{div}(D_\alpha(\nabla u)\nabla u)$$

with initial condition $u(\cdot, 0) = f$ and homogeneous Neumann boundary conditions. On the other hand, the Euler–Lagrange equation can be rewritten as

$$\frac{u - f}{\lambda} = \operatorname{div}(D_\alpha(\nabla u)\nabla u).$$

7

This can be interpreted as a fully implicit time discretization of the diffusion equation

$$\partial_t u = \operatorname{div}(D_\alpha(\nabla u)\nabla u) \tag{16}$$

with initial image $u(\cdot, 0) = f$ and homogeneous Neumann boundary conditions. In our numerical examples, we will restrict our attention to a time explicit scheme to solve (16). Of course, such a scheme, in contrast to an implicit discretization of (16) with time step size λ, will not lead to a minimizer of (13). For details see [13]. However, we propose this scheme as an alternative denoising technique which leads to very good results which we were not able to improve by adding e.g. the reaction term.

More precisely, in our experiments we will solve (16) by finite difference methods with an explicit Euler discretization of the time derivative. Here we have to be very careful with the discretization of the spatial derivatives to obtain comparable results with those from quadratic programming. We apply the first order difference filters proposed in [12, 17]:

$$
\tilde{H}_0 := \frac{1}{16}
\begin{pmatrix}
13 & 3 & & & & \\
3 & 10 & 3 & & & \\
& & \ddots & & & \\
& & & 3 & 10 & 3 \\
& & & & 3 & 13
\end{pmatrix}, \quad
\tilde{H}_1 := \frac{1}{2}
\begin{pmatrix}
-1 & 1 & & & & \\
-1 & 0 & 1 & & & \\
& & \ddots & & & \\
& & & -1 & 0 & 1 \\
& & & & -1 & 1
\end{pmatrix}.
$$

These filters which result in 5×5 spatial masks in (16) are optimized (among all 5×5 masks) with respect to rotation invariance and lead to sharp edges. Again, if we suppose pixel distances $h_x, h_y \neq 1$, we have to multiply $\tilde{H}_1$ by $1/h_x$ to obtain a consistent discretization of the derivative in x-direction. Moreover, to avoid possible checkerboard effects we use the following method suggested in [12, Section 9.5]: the fully discrete scheme of (16) reads

$$u^{n+1} = (I_N + \tau M)u^n$$

with discretization Mu of $\operatorname{div}(D_\alpha(\nabla u)\nabla u)$ and time step size τ. This scheme was modified in [12] to

$$u^{n+1} = \left(I_N + \tau M + \nu(\tilde{I} - I_N) \right) u^n$$

with a small weight parameter ν and the usual Neumann boundary modifications. The low pass filter $\tilde{I}$ is represented by the 5-band Toeplitz matrix with band $\frac{1}{16}(-1, 4, 10, 4, -1)$ and is a discretization of the identity filter I_N of consistency order 4. The idea behind the additional term $\tilde{I} - I_N$ is to add a filter, which on the one hand does no harm to the consistency of the scheme, but on the other hand eliminates the checkerboard effects. For further explanation and numerical examples demonstrating the influence of the additional term, see [12, Section 9.5].

As in the previous subsection, we can also handle linearly transformed rectangles by solving (16) with the diffusion tensor

$$D_\alpha(\nabla u) := A^{\mathrm{T}} R(\alpha)^{\mathrm{T}} \begin{pmatrix} \psi(\langle e_\alpha, A \nabla u \rangle) & 0 \\ 0 & \psi(\langle e_\alpha^\perp, A \nabla u \rangle) \end{pmatrix} R(\alpha) A.$$

3   Shape preservation

The functional (3) with constant rotation $\alpha = 0$ can be considered as a special case of the more general functional

$$\frac{1}{2} \int_\Omega (f - u)^2 \, d\mathbf{x} + \lambda \int_\Omega \varphi(\nabla u) \, d\mathbf{x} \tag{17}$$

with $\Omega \subset \mathbb{R}^m$ and with a finite *gauge function* $\varphi : \mathbb{R}^m \to \mathbb{R}$, i.e., a positively homogeneous, convex function which satisfies $\varphi(0) = 0$ and $\varphi(x) > 0$ for $x \neq 0$. Since φ is positively homogeneous and convex, it follows that φ is sublinear which means that it fulfills in addition $\varphi(x + y) \leq \varphi(x) + \varphi(y)$. Moreover, since φ is finite and positively homogeneous, its conjugate function φ^* is the indicator function of a closed bounded convex set C_φ containing the origin, called *Wulff shape* of φ, which is given as follows:

$$C_\varphi := \{ x \in \mathbb{R}^m : \langle x, y \rangle \leq \varphi(y) \quad \forall y \in \mathbb{R}^m \}.$$

Since φ is lower semicontinuous (lsc), we have that $\varphi^{**} = \varphi$ so that φ is the support function of C_φ, i.e.,

$$\varphi(x) = \sup_{y \in C_\varphi} \langle x, y \rangle. \tag{18}$$

The *polar function* φ^o of a gauge function φ is defined by

$$\varphi^o(x) := \sup_{y \neq 0} \frac{\langle x, y \rangle}{\varphi(y)} \tag{19}$$

and is a gauge function, too. Then C_φ can be also rewritten as

$$C_\varphi := \{ x \in \mathbb{R}^m : \varphi^o(x) \leq 1 \}.$$

For example, we have for $\varphi(x) := \|x\|_1$ that $C_\varphi = \{ x \in \mathbb{R}^m : \|x\|_\infty \leq 1 \}$. For $m = 2$, the set C_φ represents the square centered at the origin with sides of length 2 parallel to the x- and y-axis.

In [6], it was proven for the continuous setting that if f is the characteristic function of the Wulff shape of φ, i.e., $f = 1_{C_\varphi}$ and λ is small enough, then the minimizer $\hat{u}$ of (17) preserves this form which means that $\hat{u} = c \, 1_{C_\varphi}$ with an appropriate constant $c > 0$. Moreover, it was shown that the minimizer

of (17) with $\varphi(x) = \|x\|_1$ also preserves rectangles with sides parallel to the axes.

For a vector $U := \left(U_{i+kN}\right)_{i=0,k=0}^{N-1,m-1} \in \mathbb{R}^{mN}$, we set $U_i := \left(U_{i+kN}\right)_{k=0}^{m-1} \in \mathbb{R}^m$. We are interested in the discrete counterpart to (17) given by

$$\frac{1}{2}\|f - u\|^2 + \lambda\|\left(\varphi\left((Lu)_i\right)\right)_{i=0}^{N-1}\|_1 \tag{20}$$

with a finite lsc gauge function $\varphi : \mathbb{R}^m \to \mathbb{R}$ and a matrix $L \in \mathbb{R}^{mN,N}$. The typical example for $L \in \mathbb{R}^{2N,N}$ in the previous section was $L = (D_x^{\mathrm{T}} \; D_y^{\mathrm{T}})^{\mathrm{T}}$. The penalizing term

$$J(u) := \|\left(\varphi\left((Lu)_i\right)\right)_{i=0}^{N-1}\|_1$$

is also a finite lsc gauge function and is therefore the support function of

$$C_J := \{v \in \mathbb{R}^N : \langle v, w \rangle \leq J(w) \quad \forall w \in \mathbb{R}^N\}. \tag{21}$$

It is well-known, cf. [10, 4], that the unique minimizer of (20) is given by

$$\hat{u} = f - \lambda\hat{v}, \tag{22}$$

where $\lambda\hat{v}$ is the orthogonal projection of f onto $C_{\lambda J}$. Using that $\lambda\hat{v} \in C_{\lambda J}$ if and only if $\hat{v} \in C_J$ we have that $\hat{v} := \arg\min_{v \in C_J} \|f - \lambda v\|_2$. Concerning orthogonal projection the following lemma will be useful, see [6] for the continuous setting with $L = \nabla$.

Lemma 3.1. *Let J be the support function of the closed convex set C_J defined by (21). Then we have that $\lambda\hat{v}$ is the orthogonal projection of f onto $C_{\lambda J}$ if and only if $\hat{v} \in C_J$ and*

$$\langle f - \lambda\hat{v}, \hat{v} \rangle = J(f - \lambda\hat{v}). \tag{23}$$

We add the brief proof for our discrete approach.

Proof: Let $\lambda\hat{v}$ be the orthogonal projection of f onto $C_{\lambda J}$. Then we obtain by the projection theorem that

$$\langle f - \lambda\hat{v}, y - \lambda\hat{v} \rangle \leq 0 \quad \forall y \in C_{\lambda J}.$$

Together with (18) this implies that

$$\lambda J(f - \lambda\hat{v}) = \sup_{y \in C_{\lambda J}} \langle y, f - \lambda\hat{v} \rangle = \langle f - \lambda\hat{v}, \lambda\hat{v} \rangle$$

and hence (23).

Assume conversely that $\hat{v} \in C_J$ fulfills (23). Then $\lambda\hat{v} \in C_{\lambda J}$ and we obtain by the definition of $C_{\lambda J}$ that for all $y \in C_{\lambda J}$

$$
\begin{aligned}
\langle f - \lambda\hat{v}, y \rangle &\leq \lambda J(f - \lambda\hat{v}) \\
0 &\leq -\langle f - \lambda\hat{v}, y \rangle + \lambda J(f - \lambda\hat{v}) = \langle f - \lambda\hat{v}, \lambda\hat{v} - y \rangle
\end{aligned}
$$

which yields by the projection theorem that $\lambda\hat{v}$ is the orthogonal projection of f onto $C_{\lambda J}$. $\qquad\square$

To determine $\hat{v}$ we need a more manageable form of C_J which is given in the following lemma.

Lemma 3.2. *The set C_J defined by (21) coincides with*

$$
\widetilde{C}_J := \{v = L^{\mathrm{T}}V : \|\big(\varphi^o(V_i)\big)_{i=0}^{N-1}\|_\infty \leq 1\}. \tag{24}
$$

Note that the condition on V is equivalent to $V_i \in C_\varphi$ for all $i = 0, \ldots, N-1$.

Proof: 1. First we see that $v \in C_J$ must fulfill $v \in \mathcal{R}(L^{\mathrm{T}})$ since otherwise we can use the orthogonal decomposition $v = v_0 + v_1$ with $v_0 \in \mathcal{N}(L)$, $v_0 \neq 0$ and $v_1 \in \mathcal{R}(L^{\mathrm{T}})$ to obtain the contradiction

$$
\langle v, v_0 \rangle = \langle v_0 + v_1, v_0 \rangle = \|v_0\|_2^2 \leq J(v_0) = 0.
$$

Thus,

$$
C_J = \{v = L^{\mathrm{T}}V : \langle V, Lw \rangle \leq \|\big(\varphi\left((Lw)_i\right)\big)_{i=0}^{N-1}\|_1 \quad \forall w \in \mathbb{R}^N\}. \tag{25}
$$

2. Let $v \in \widetilde{C}_J$. By definition of φ^o in (19), we see that

$$
\langle V_i, W_i \rangle \leq \varphi^o(V_i)\varphi(W_i) \quad \forall W_i \in \mathbb{R}^m
$$

so that we get for $W_i = (Lw)_i$ by assumption

$$
\langle V, Lw \rangle = \sum_{i=0}^{N-1} \langle V_i, (Lw)_i \rangle \leq \sum_{i=0}^{N-1} \varphi^o(V_i)\varphi((Lw)_i) \leq \sum_{i=0}^{N-1} \varphi((Lw)_i).
$$

By (25) this yields $v \in C_J$.

3. Conversely, let $v = L^{\mathrm{T}}V \in C_J$. We have to show that there exists a $\tilde{V}$ such that $v = L^{\mathrm{T}}\tilde{V}$ and $\|\big(\varphi^o(\tilde{V}_i)\big)_{i=0}^{N-1}\|_\infty \leq 1$.

The functional $l_V(Lw) := \langle V, Lw \rangle$ is linear on $\mathcal{R}(L) \subset \mathbb{R}^{mN}$ and satisfies

$$
l_V(Lw) \leq p(Lw) \quad \forall w \in \mathbb{R}^N,
$$

where $p : \mathbb{R}^{mN} \to \mathbb{R}$ is the sublinear function $p(W) := \|\big(\varphi\left(W_i\right)\big)_{i=0}^{N-1}\|_1$. By the Hahn-Banach theorem l_V can be extended to a linear functional $L_{\tilde{V}}(W) := \langle \tilde{V}, W \rangle$ on $\mathbb{R}^{mN}$ which fulfills

$$
\langle \tilde{V}, W \rangle \leq p(W) \quad \forall W \in \mathbb{R}^{mN}. \tag{26}
$$

11

Now $L_{\tilde{V}}(Lw) = l_V(Lw)$ for all $w \in \mathbb{R}^N$, i.e., $\langle L^{\mathrm{T}}V, w \rangle = \langle L^{\mathrm{T}}\tilde{V}, w \rangle$ for all $w \in \mathbb{R}^N$ which implies that $v = L^{\mathrm{T}}V = L^{\mathrm{T}}\tilde{V}$.

Let $i_0 \in \{0, \dots, N-1\}$ so that $\varphi^o(\tilde{V}_{i_0}) = \max\{\varphi^o(\tilde{V}_i) : i = 0, \dots, N-1\}$. Then we have by (26) for all $\tilde{W}$ with $\tilde{W}_i := 0$ for $i \neq i_0$ that

$$\langle \tilde{V}, \tilde{W} \rangle = \langle \tilde{V}_{i_0}, \tilde{W}_{i_0} \rangle \leq \varphi(\tilde{W}_{i_0}), \quad \forall \tilde{W}_{i_0} \in \mathbb{R}^m,$$

$$\varphi^o(\tilde{V}_{i_0}) = \sup_{\tilde{W}_{i_0} \neq 0} \frac{\langle \tilde{V}_{i_0}, \tilde{W}_{i_0} \rangle}{\varphi(\tilde{W}_{i_0})} \leq 1$$

and we are done by assumption on $\tilde{V}_{i_0}$. $\qquad\square$

It is not clear to us how to define some kind of Wulff shapes on a discrete grid for general φ. However, we may consider rectangles on $\mathbb{Z}^2$ and ask if they are preserved by the minimizer of (20) for $\varphi(x) = \|x\|_1$ and $L = (D_x^{\mathrm{T}}\ D_y^{\mathrm{T}})^{\mathrm{T}}$. Of course this depends on the concrete choice of the difference matrices D_x and D_y. In the following, we provide an example. Similarly as in [6], we use that (22) and the Lemmas 3.1 and 3.2 imply that $\hat{u}$ is the minimizer of (20) if and only if the following relations are fulfilled:

i) $\hat{u} = f - \lambda\hat{v}$,

ii) $\hat{v} = L^{\mathrm{T}}\hat{V}$, where $\hat{V}_i \in C_\varphi$ for all $i = 0, \dots, N-1$,

iii) $J(\hat{u}) = \langle \hat{u}, \hat{v} \rangle$.

Example 3.3. *Let $\varphi(x, y) = |x| + |y|$ so that ii) is equivalent to $\|\hat{V}\|_\infty \leq 1$. Let $R := \{x_0 + 1, \cdots, x_0 + a\} \times \{y_0 + 1, \cdots, y_0 + b\}$ with $x_0, y_0 \geq 0$ and $x_0 + a, y_0 + b \leq n - 2$ and let $\mathbf{f} = 1_R$ be the image $\mathbf{f} : \{0, \dots, n-1\} \times \{0, \dots, n-1\} \to \mathbb{R}$ with $f(x, y) = 1$ if $(x, y) \in R$ and $f(x, y) = 0$ otherwise. Furthermore, we use $L := \begin{pmatrix} H_0 \otimes H_1 \\ H_1 \otimes H_0 \end{pmatrix}$, where we slightly modify the matrices in (6) by $h_0(0, 0) = h_1(0, 0) = 0$ and $h_0(n-1, n-1) = 1$, $h_1(n-1, n-1) = -1$, i.e., H_1 is the forward difference matrix with zero boundary assumptions and H_0 is the corresponding smoothing matrix. Consider $\hat{V} = \begin{pmatrix} \hat{V}^1 \\ \hat{V}^2 \end{pmatrix}$ defined*

by

$$\hat{V}^1(x,y) \;=\; \begin{cases} (-1)^{y_0-y} & x \in I_1, \ y \in J_1, \\ 1 & x \in I_1, \ y \in J_2, \\ (-1)^{y_0+b-y} & x \in I_1, \ y \in J_3, \\ (-1)^{y_0-y}\left(1 - \frac{2(x-x_0)}{a}\right) & x \in I_2, \ y \in J_1, \\ 1 - \frac{2(x-x_0)}{a} & x \in I_2, \ y \in J_2, \\ (-1)^{y_0+b-y}\left(1 - \frac{2(x-x_0)}{a}\right) & x \in I_2, \ y \in J_3, \\ (-1)^{y_0-y+1} & x \in I_3, \ y \in J_1, \\ -1 & x \in I_3, \ y \in J_2, \\ (-1)^{y_0+b-y+1} & x \in I_3, \ y \in J_3, \end{cases}$$

and

$$\hat{V}^2(x,y) \;=\; \begin{cases} (-1)^{x_0-x} & x \in I_1, \ y \in J_1, \\ 1 & x \in I_2, \ y \in J_1, \\ (-1)^{x_0+a-x} & x \in I_3, \ y \in J_1, \\ (-1)^{x_0-x}\left(1 - \frac{2(y-y_0)}{b}\right) & x \in I_1, \ y \in J_2, \\ 1 - \frac{2(y-y_0)}{b} & x \in I_2, \ y \in J_2, \\ (-1)^{x_0+a-x}\left(1 - \frac{2(y-y_0)}{b}\right) & x \in I_3, \ y \in J_2, \\ (-1)^{x_0-x+1} & x \in I_1, \ y \in J_3, \\ -1 & x \in I_2, \ y \in J_3, \\ (-1)^{x_0+a-x+1} & x \in I_3, \ y \in J_3, \end{cases}$$

where $I_1 := \{0, \cdots, x_0\}$, $I_2 := \{x_0+1, \cdots, x_0+a\}$, $I_3 := \{x_0+a+1, \cdots, n-1\}$,
and $J_1 := \{0, \cdots, y_0\}$, $J_2 := \{y_0+1, \cdots, y_0+b\}$, $J_3 := \{y_0+b+1, \cdots, n-1\}$.
Of course $\hat{V}$ fulfills ii). Further we see that $\hat{v} = L^T\hat{V} = \left(\frac{2}{a} + \frac{2}{b}\right)1_R$. Then
we obtain for

$$\hat{u} = f - \lambda\hat{v} = (1 - \frac{2(a+b)}{ab}\lambda)\,1_R = c\,1_R \tag{27}$$

that

$$J(\hat{u}) = 2(a-1)c + 4\frac{c}{2} + 2(b-1)c + 4\frac{c}{2} = 2(a+b)\,c = \langle\hat{u},\hat{v}\rangle.$$

Thus, for $\lambda \leq \frac{ab}{2(a+b)}$, the function $\hat{u}$ in (27) is the minimizer of (20).

4   Numerical examples

Finally, we present numerical examples. All programs were written in MAT-LAB. Further, we have used the a primal-dual predictor-corrector interior

point method implemented in the software package MOSEK [2] to solve the quadratic problem with linear constraints (11). To visualize the images we have used the MATLAB `imagesc` routine which incorporates an affine gray value scaling to use the full gray value map. The parameters are chosen with respect to the best visuable results.

The first two examples in Figs. 2 and 3 demonstrate the denoising of rotated rectangles and parallelograms obtained by applying a shear matrix with $s = 0.92$. In both examples it suffices to perform only one angle adaptation and restoration step. We show the results for the restoration by quadratic programming and anisotropic diffusion. We observe a slight smoothing of the edges in Fig. 3 in the second approach. This can be reduced by chosing a smaller parameter ν with the disadvantage that checkerboard effects become visible. Since the images are depicted with the affine gray value scaling of the MATLAB `imagesc` routine we remark that the computed minimal/maximal gray values are given as follows:

Fig. 2 bottom left: $-3.1 \cdot 10^{-7}$, 220.3,

Fig. 2 bottom right: 7, 223.2,

Fig. 3 bottom left: -0.5, 249.5,

Fig. 3 bottom right: -7.8, 252.4.

The original gray values ranged from 0 to 255.

The third example in Fig. 4 depicts the cartoon extraction from a real-world image which was also presented in [3]. For comparison we found it useful to take the same image as in [3]. We have iterated our two steps three times.

Finally, we added an image from [1] which is often used as a challenging example for the comparison of denoising algorithms, see, e.g., [16, 18]. We present the results after 4 iterations with quadratic programming and anisotropic diffusion. Moreover we show versions without and with rescaling to get an impression of the loss of contrast. Here the minimal/maximal gray values are 106.4, 163.6 for quadratic programming and 95.1, 173.3 for anisotropic diffusion.

5   Summary and Conclusions

We have proposed a method for the restoration of rectangular shapes contaminated with heavy noise which avoids the round-off effects at vertices produced by known edge-preserving denoising techniques. As in a paper of Berkels et al. our procedure approximates the rotation angle of the shapes in a first step and uses this information in a second step to denoise the image without destroying vertices. Our angle adaptation uses only first order derivatives of the linearly smoothed image and requires to solve of a linear system of equations. For the second step, we have proposed two different methods, namely quadratic programming and an anisotropic diffusion process with the diffusion tensor adapted to the rotation angle.
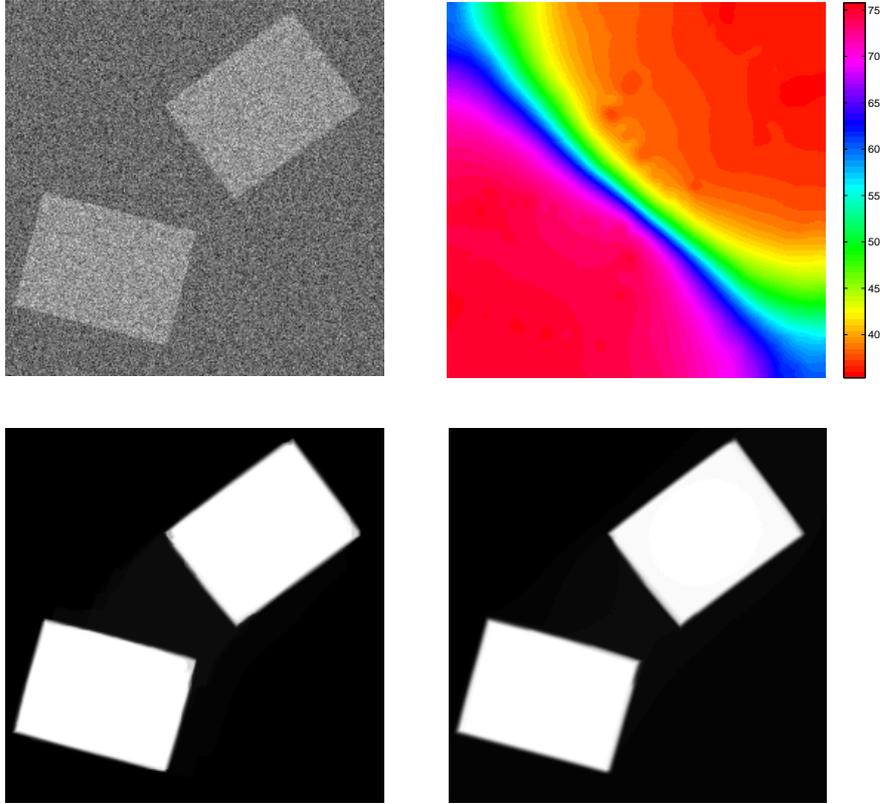
Figure 2: Top left: noisy image with standard deviation 150. Top right: smoothed rotation angles used to denoise the image with parameters $\mu = 400000$ and $\sigma = 2.5$. Bottom left: denoised image by quadratic programming with $\lambda = 800$ and $\nu = 0.1$. Bottom right: denoised image by anisotropic diffusion with $\varepsilon^2 = 0.001$, time step size $\tau = 0.1$, number of iterations itmax $= 8000$ and $\nu = 0.001$.

So far, the algorithm works for rectangles and linearly transformed rectangles, where the linear transform has to be known in advance. We have demonstrated this also numerically for sheared rectangles (parallelograms) with fixed shear parameter. Our ongoing work aims at adapting the shear parameter, too. Moreover, we want to generalize the approach to the restoration of arbitrary (polygonal) shapes. This will incorporate the application of more sophisticated corner detectors and their inclusion into the diffusion tensor of a diffusion equation or an appropriate functional.

References

[1] *MegaWave*. http://www.cmla.ens-cachan.fr/Cmla/Megawave.
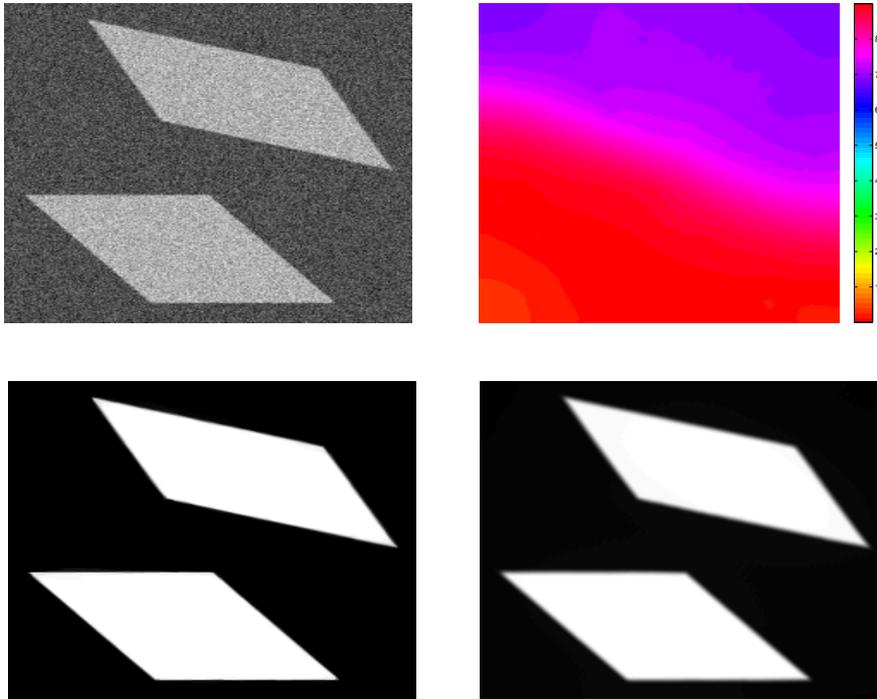
Figure 3: Top left: noisy image containing sheared rectangles with fixed shear parameter $s = 0.92$ and white Gaussian noise of standard deviation 50. Top right: smoothed rotation angles used to denoise the image with parameters $\mu = 300000$ and $\sigma = 1.5$. Bottom left: denoised image by quadratic programming with $\lambda = 150$ and $\nu = 0.04$. Bottom right: denoised image by anisotropic diffusion with $\varepsilon^2 = 0.001$, time step size $\tau = 0.1$, number of iterations itmax $= 1800$ and $\nu = 0.01$.

[2] *The MOSEK Optimization Toolbox.* http://www.mosek.com.

[3] B. Berkels, M. Burger, M. Droske, O. Nemitz, and M. Rumpf. Cartoon extraction based on anisotropic image classification, vision, modeling and visualization proceedings. In *Vision, Modeling and Visualization.* Springer, accepted.

[4] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, (20):89–97, 2004.

[5] A. Chambolle. Total variation minimization and a class of binary MRF models. In A. Rangarajan, B. C. Vemuri, and A. L. Yuille, editors, *Energy Minimization Methods in Computer Vision and Pattern Recog-*

nition, EMMCVPR, volume 3757 of *Lecture Notes in Computer Science*. Springer, 2005.

[6] S. Esedoglu and S. Osher. Decomposition of images by anisotropic Rudin-Osher-Fatemi model. *Communications in Pure and Applied Mathematics*, 57(12):1609–1626, 2004.

[7] W. Hintermüller and W. Kunisch. Total bounded variation regularization as a bilaterally constrained optimization problem. *SIAM J. Appl. Math.*, 4(64):1311–1333, 2004.

[8] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, second edition, 1999.

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

[10] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer, Berlin, 1998.

[11] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica A*, 60:259–268, 1992.

[12] H. Scharr. *Optimal Operators in Digital Image Processing*. PhD thesis, Interdisciplinary Center for Scientific Computing, Ruprecht-Karls-Universität Heidelberg, 2000.

[13] O. Scherzer and J. Weickert. Relations between regularization and diffusion filtering. *Journal of Mathematical Imaging and Vision*, 12:43–63, 2000.

[14] G. Steidl. A note on the dual treatment of higher order regularization functionals. *Computing*, 76:135 – 148, 2006.

[15] J. Weickert. Anisotropic diffusion filters for image processing based quality control. In A. Fasano and M. Primicerio, editors, *Proc. Seventh European Conference on Mathematics in Industry*, pages 355–362. Teubner, Stuttgart, 1994.

[16] J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart, 1998.

[17] J. Weickert and H. Scharr. A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance. *Journal of Visual Communication and Image Representation*, 13(1/2):103–118, 2002.

[18] M. Welk, G. Steidl, and Weickert. Locally analytic schemes: a link between diffusion filtering and wavelet shrinkage. *Applied and Computational Harmonic Analysis*, to appear.

[19] G. Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*, volume 27 of *Applications of Mathematics*. Springer, Berlin, 1995.
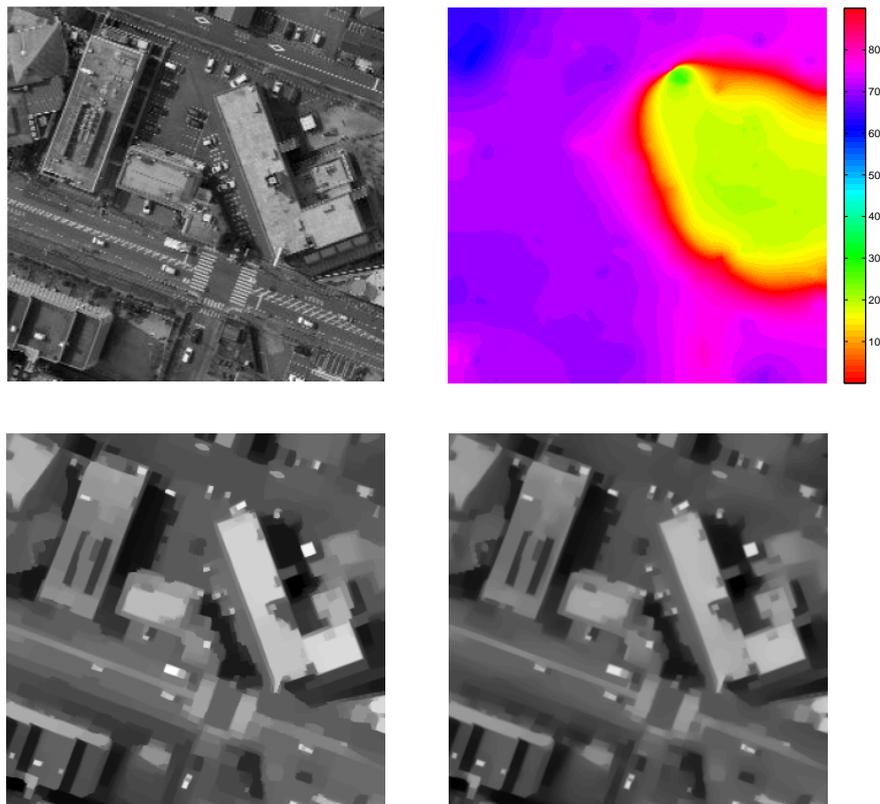
Figure 4: Top left: original image of a city area as presented in [3]. Top right: smoothed rotation angles used to obtain the final image at bottom left with parameters $\mu = 20000$ and $\sigma = 0.8$. The smoothed rotation angles belonging to the anisotropic diffusion at the bottom right look very similar. Bottom left: cartoon generated by quadratic programming with $\lambda = 50$ and $\nu = 0$. Bottom right: cartoon generated by anisotropic diffusion with $\varepsilon^2 = 0.1$, time step size $\tau = 0.05$, number of iterations itmax $= 700$ and $\nu = \frac{1}{200}$.
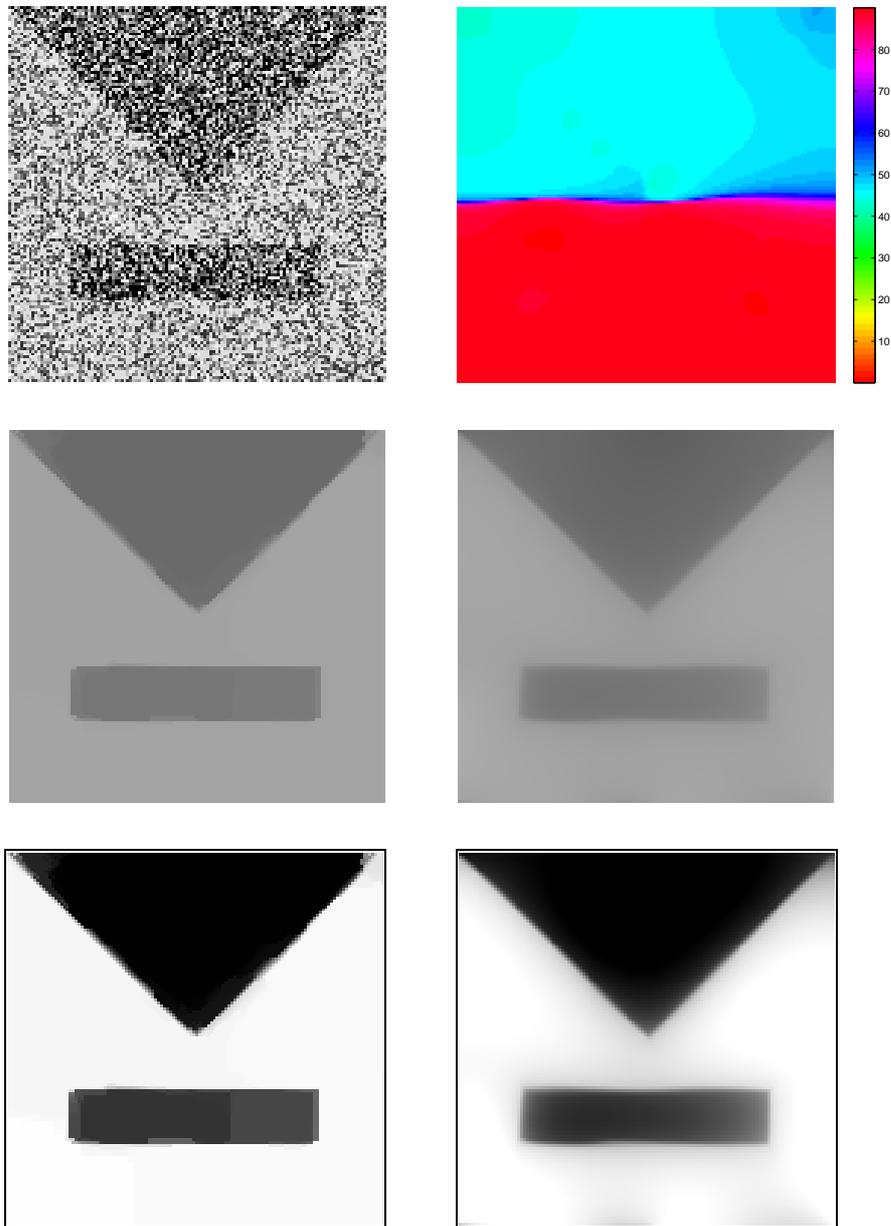
Figure 5: Top left: noisy image. Top right: smoothed rotation angles used to generate the final image on the left with parameters $\mu = 10000$ and $\sigma = 1.5$. The smoothed rotation angles belonging to the anisotropic diffusion look very similar. Middle left: denoised image by quadratic programming with $\lambda = 150$ and $\nu = 0.1$ without rescaling. Middle right: denoised image by anisotropic diffusion with $\varepsilon^2 = 0.5$, time step size $\tau = 0.1$, number of iterations itmax $= 1700$ and $\nu = \frac{1}{300}$ without rescaling. Bottom: same as in the middle with identical rescaling.