Diffusion tensors for processing sheared and rotated rectangles

G. Steidl and T. Teuber [*]

Abstract

Image restoration and simplification methods that respect important features such as edges play a fundamental role in digital image processing. However, known edge-preserving methods like common nonlinear diffusion methods tend to round vertices for large diffusion times. In this paper, we adapt the diffusion tensor for anisotropic diffusion to avoid this effects in images containing rotated and sheared rectangles, respectively. In this context we propose a new method for estimating rotation angles and shear parameters based on the so-called structure tensor. Further, we show how the knowledge of appropriate diffusion tensors can be used in variational models. Numerical examples including orientation estimation, denoising and segmentation demonstrate the good performance of our methods.

1 Introduction

Given a possibly noisy image $f : \Omega \to \mathbb{R}$, one method to denoise and simplify the image consists in assuming f to be the initial state of a diffusion process

$$\begin{aligned} \partial_t u &= \operatorname{div}\left(D(\nabla u_\sigma)\nabla u \right) && \text{diffusion equation,} \qquad (1)\\ \left(D(\nabla u_\sigma)\nabla u \right)^{\mathrm{T}} n &= 0 && \text{Neumann boundary conditions,}\\ u(\cdot,0) &= f && \text{initial condition} \end{aligned}$$

and taking the diffused image u after an appropriate time as the final image. Here n denotes the outward normal to the image boundary,

$$u_\sigma := K_\sigma * u$$

is the convolved image with a Gaussian K_σ of standard deviation σ and $D(\nabla u_\sigma(x)) \in \mathbb{R}^{2,2}$ is the so-called *diffusion tensor*. We remark that the convolution with the Gaussian was recently also replaced by wavelet-based smoothing procedures, see, e.g., [20]. The diffusion tensor plays a key role in the diffusion process since it steers the direction of the *flux*

[*] G. Steidl and T. Teuber are with the Dept. of Mathematics and Computer Science, University of Mannheim, 68131 Mannheim, Germany (email: steidl@math.uni-mannheim.de, tteuber@kiwi.math.uni-mannheim.de).

$j := -D(\nabla u_\sigma)\nabla u$. If $D(\nabla u_\sigma) = \mathrm{diag}\big((g(|\nabla u_\sigma|), g(|\nabla u_\sigma|))\big)$ is a diagonal matrix with identical diagonal entries $g(|\nabla u_\sigma|)$, then the diffusion equation becomes

$$\partial_t u \;=\; \mathrm{div}(g(|\nabla u_\sigma|)\,\nabla u). \tag{2}$$

In this case, the flux is only influenced in gradient direction. More precisely, if the diffusivity $g : \mathbb{R}_{\geq 0} \to [0, 1]$ is a decreasing function with $g(0) = 1$ as, e.g., the *Perona–Malik diffusivity* $g(|x|) := 1/(1 + \frac{|x|^2}{\gamma^2})$ introduced in [19], then the flux becomes smaller for high absolute gradient values. Thus, the diffusion is lowered at important image features such as edges which leads to edge preserving methods. However, since the diffusion in gradient direction at edges is lowered, noisy pixels survive at edges for a long time. Of course, if we diffuse sufficiently long, these pixels are smoothed too, but the denoising result is not satisfactory at vertices which are rounded, see Fig. 1 bottom left and middle. At least the first effect can be avoided by using the so-called *edge-enhanced diffusion* (EED), see [28], which incorporates the more sophisticated diffusion tensor $D(\nabla u_\sigma)$ described in the next section. This diffusion tensor, which is a symmetric 2×2 matrix now, allows a diffusion perpendicular to the gradient direction and thus cleans the noisy pixels along edges. However, we are still confronted with rounded vertices as shown in Fig. 1 bottom right.

Besides the described diffusion methods there exists a large number of other denoising techniques. Energy minimization and adaptive filtering are closely related to PDE methods and we want to comment on some of these approaches. The Rudin-Osher-Fatemi variational method [21] shows the same problem at vertices if the regularization parameter must be chosen rather large to eliminate the noise, see Fig. 1 top right and also the results in [4]. Further, we observed that bilateral filters [26, 18] cannot cope with this large amount of noise. In a series of papers, Kimmel, Sochen and others suggested restoration techniques within the Beltrami framework. The corresponding smoothing with the so-called 'short-time Beltrami kernel' proposed in [25] differs from the bilateral filters in the fact that it uses geodetic distances on the image manifold while the bilateral kernel applies Euclidian distances.

Another approach for adapting a PDE to the local geometry of an image can be found in [27]. Here, the author estimates the local structure by the so-called 'structure tensor' originally introduced in [13] and then performs the diffusion along integral curves deduced from those structure tensors. It was shown that the solution of the regarded PDE can be approximated by successively convolving the initial image with one-dimensional Gaussians along these integral curves which leads to "curved" filtering. This method is more flexible with respect to different shapes than the approach presented in this paper. However, we observed again rounding artefacts at corners in the presence of heavy noise.

We obtained much better results by the recently proposed nonlocal means (NL means) at least if we apply the method iteratively. Among the meanwhile rich literature on NL means we refer to [5] since we have used the corresponding MATLAB software [16] to produce the restored image in Fig. 5 bottom right. The result is only slightly worse in comparison with our methods.

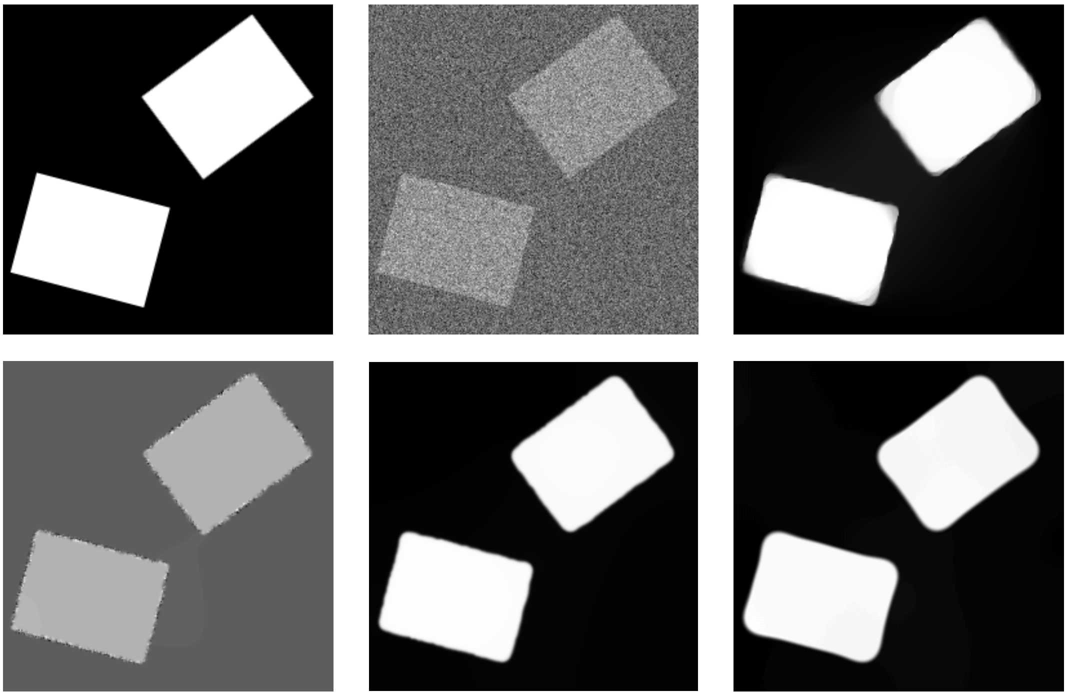

Figure 1: Top left: original image. Top middle: noisy image corrupted by heavy white Gaussian noise of standard deviation 150. Top right: denoised image by ROF with regularization parameter $\lambda = 1000$. One can improve the result of the ROF method at the slightly bumpy edges by choosing a larger regularization parameter, but then the vertices become more rounded. Bottom left/middle : denoised images by the nonlinear diffusion (2) with $\sigma = 3$ and stopping times $t = 400$ (left) and $t = 3000$ (middle). For the moderate diffusion time $t = 400$ noisy pixels survive at edges which is not the case for EED (bottom right). However, both methods suffer from rounding artefacts at vertices. Bottom right: denoised image by EED with $\sigma = 2.5$ and $t = 400$.

For the special case of rotated rectangles, the topic of vertex preserving image simplification was addressed by Berkels et al. [4] who suggested, based on results of Osher and Esedoglu [12], a regularization approach. This leads to an alternating two-step algorithm which computes in the first step the rotation angles of the rectangles by minimizing a functional that contains first and second order derivatives of the rotation angles. A simpler regularization method for finding the rotation angles which includes only first order derivatives was suggested in [24].

In this paper, we focus on diffusion methods. We develop a new method for estimating the rotation angles and the shear parameters of rotated and sheared rectangles in images by adapting the structure tensor. Based on these estimates we propose a modification of the diffusion tensor in anisotropic PDEs so that rounding artefacts at vertices are avoided. The numerical computations are based on finite difference methods. The corresponding diffusion process shows very good results for images containing rotated rectangles or sheared rectangles.

Our paper is organized as follows: In the next section, we present our diffusion models. We start by recalling EED in Subsection 2.1. Then we modify the approach in Subsections 2.2 and 2.3 such that vertices of rotated rectangles and of sheared rectangles are preserved. In particular, we present a method for determining rotation angles and shear parameters. In Section 3, we sketch how our new diffusion tensors can be applied to establish a regularization model for vertex preservation. Section 4 contains discretization issues. The good performance of our models is demonstrated in Section 5. Our numerical examples include grouping by orientation estimation, denoising and segmentation. Finally, a summary and conclusions are given in Section 6.

2 Anisotropic diffusion model

In this section, we present adaptations of the diffusion tensor for vertex preservation of rotated and sheared rectangles. We start by recalling EED.

2.1 Structure tensor and diffusion tensor

EED applies the structure tensor concept to define useful diffusion tensors. For any $x \in \Omega$, we consider $\nabla u_\sigma(x) = \nabla u_\sigma = |\nabla u_\sigma|(c,s)^{\mathrm{T}}$, where $c := \cos\alpha$ and $s := \sin\alpha$ and

$$\alpha := \begin{cases} \arccos \frac{\partial_x u_\sigma}{|\nabla u_\sigma|} & \text{if } \partial_y u_\sigma \geq 0, \\ 2\pi - \arccos \frac{\partial_x u_\sigma}{|\nabla u_\sigma|} & \text{if } \partial_y u_\sigma < 0 \end{cases} \tag{3}$$

denotes the angle between ∇u_σ and the x-axis. We set $\alpha := 0$ if $|\nabla u_\sigma| = 0$. Let

$$J_0(\nabla u_\sigma) \;:=\; \nabla u_\sigma \nabla u_\sigma^{\mathrm{T}} \;=\; |\nabla u_\sigma|^2 \begin{pmatrix} c^2 & c\,s \\ c\,s & s^2 \end{pmatrix}. \tag{4}$$

Obviously, the entries of J_0 do not distinguish between left and right directions, i.e., J_0 is the same for $\alpha \bmod \pi$. By definition, J_0 is a rank-1 matrix with spectral decomposition

$$J_0(\nabla u_\sigma) \;=\; Q \begin{pmatrix} |\nabla u_\sigma|^2 & 0 \\ 0 & 0 \end{pmatrix} Q^{\mathrm{T}} \;=\; \begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} |\nabla u_\sigma|^2 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} c & s \\ -s & c \end{pmatrix}.$$

In [13], Förstner and Gülch defined the *structure tensor* J_ρ, $\rho > 0$, as smoothed variant of J_0 by

$$J_\rho(\nabla u_\sigma) \;:=\; K_\rho * J_0(\nabla u_\sigma) \;=\; V \begin{pmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{pmatrix} V^{\mathrm{T}},$$

where the last equality gives the spectral decomposition of J_ρ. Now EED, see [28], uses the diffusion tensor

$$D(\nabla u_\sigma) \;:=\; g(J_0(\nabla u_\sigma)) \;=\; Q \begin{pmatrix} g(|\nabla u_\sigma|^2) & 0 \\ 0 & 1 \end{pmatrix} Q^{\mathrm{T}}.$$

Alternatively, one could work with $g(J_\rho(\nabla u_\sigma)) = V \begin{pmatrix} g(\mu_1) & 0 \\ 0 & g(\mu_2) \end{pmatrix} V^{\mathrm{T}}$. For $D(\nabla u_\sigma)$ the negative flux becomes

$$D(\nabla u_\sigma)\nabla u = g(|\nabla u_\sigma|^2)\frac{\nabla u_\sigma^{\mathrm{T}} \nabla u}{|\nabla u_\sigma|^2}\,\nabla u_\sigma \;+\; 1\,\frac{(\nabla u_\sigma^\perp)^{\mathrm{T}}\nabla u}{|\nabla u_\sigma|^2}\,\nabla u_\sigma^\perp$$

and the second summand indicates that we allow a flux component perpendicular to the gradient which is responsible for the cleaned edges in Fig. 1 bottom right in contrast to Fig. 1 bottom left.

2.2 Adapting the diffusion tensor to rotated rectangles

If we know that an image consists only of rotated rectangles we can use this knowledge to modify the diffusion tensor. Having defined the angles α as in (3), we set $c_2 := \cos(2\alpha)$, $s_2 := \sin(2\alpha)$ such that $|\nabla u_\sigma|(c_2, s_2)^{\mathrm{T}}$ is the gradient ∇u_σ rotated by its angle α. Then, instead of computing (4), we adapt J_0 as follows:

$$J_0\big(|\nabla u_\sigma| \binom{c_2}{s_2}\big) \;:=\; |\nabla u_\sigma|^2 \begin{pmatrix} c_2^2 & c_2\, s_2 \\ c_2\, s_2 & s_2^2 \end{pmatrix}.$$

Due to the period π of $\cos(2\cdot)$ and $\sin(2\cdot)$, the new matrix $J_0\big(|\nabla u_\sigma|(c_2, s_2)^{\mathrm{T}}\big)$ can only distinguish between angles mod $\frac{\pi}{2}$ and its smoothing should nicely relate the vertices to the corresponding edges. To this end, we compute

$$J_\rho\big(|\nabla u_\sigma| \binom{c_2}{s_2}\big) \;:=\; K_\rho * J_0\big(|\nabla u_\sigma| \binom{c_2}{s_2}\big) \;=\; V_2 \begin{pmatrix} \mu_{2,1} & 0 \\ 0 & \mu_{2,2} \end{pmatrix} V_2^{\mathrm{T}}. \tag{5}$$

We assume that the columns of matrices V_2 are ordered in such a way that $\mu_{2,1} \ge \mu_{2,2}$. Next, we reverse the angle doubling process. For each image point, we compute the angle $\alpha_{2,\rho}$ of the eigenvector belonging to $\mu_{2,1}$ and take $\alpha_{2,\rho}/2$ as smooth approximation of the rotation angle. Finally, we set $c_\rho := \cos(\alpha_{2,\rho}/2)$, $s_\rho := \sin(\alpha_{2,\rho}/2)$ and define the diffusion tensor by

$$D(\nabla u_\sigma) \;:=\; R^{\mathrm{T}} \begin{pmatrix} g((R\,\nabla u_\sigma)_1) & 0 \\ 0 & g((R\,\nabla u_\sigma)_2) \end{pmatrix} R, \quad R = R(\rho) := \begin{pmatrix} c_\rho & s_\rho \\ -s_\rho & c_\rho \end{pmatrix}. \tag{6}$$

Fig. 2 shows the angle smoothing process in contrast to the method proposed in [24]. The local character of the angle smoothing via the structure tensor (5) is obvious.

Remark. Similarly, we can also consider rotated and sheared rectangles, where we know the shear matrices

$$S = S(\zeta) := \begin{pmatrix} 1 & 0 \\ \zeta & 1 \end{pmatrix} \tag{7}$$

related to the different parallelograms in advance. More precisely, we suppose that we know the matrices $S^{-1} \in \mathbb{R}^{2,2}$ which have transformed the gradients of rectangles into the gradients of parallelograms contained in our image, see Fig. 3. Since S transforms the

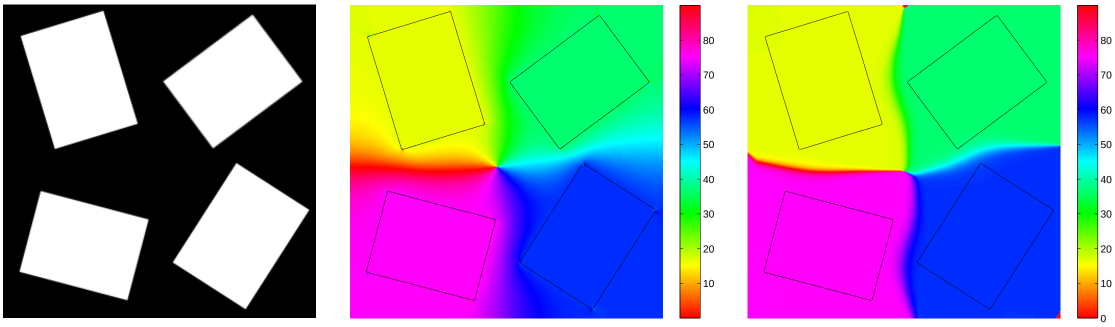

Figure 2: Left: original image of different rotated rectangles. Middle: estimated rotation angles by the variational angle adaptation method proposed in [24] with parameters $\sigma = 0.5$ and $\mu = 10000$. Right: estimated rotation angles by our new structure tensor method with parameters $\sigma = 0.5$ and $\rho = 13$.

gradients of the image with parallelograms back to the gradients of an image with rotated rectangles we can process such images by slightly modifying the diffusion tensor. We just define α to be the angles of $S\nabla u_\sigma$ instead of ∇u_σ, compute c_ρ, s_ρ appropriately, and set the diffusion tensor to

$$D(\nabla u_\sigma) \;:=\; S^{\mathrm{T}} R^{\mathrm{T}} \begin{pmatrix} g((RS\,\nabla u_\sigma)_1) & 0 \\ 0 & g((RS\,\nabla u_\sigma)_2) \end{pmatrix} RS. \tag{8}$$

Note that the matrices RS are in general not orthogonal so that the flux is steered using the non-orthogonal directions given by their rows.

2.3 Adapting the diffusion tensor to sheared rectangles

Next, we focus on images consisting of sheared rectangles which are not rotated, but where the shear parameters ζ in (7) are not known in advance. To process such images while preserving sharp (non-orthogonal) vertices we want to incorporate an estimation of the shear parameters into the diffusion tensor. We know that if a point $x \in \Omega$ belongs to a non-horizontal edge of a sheared rectangle with shear parameter ζ, then it is likely that at this pixel

$$S(\zeta)\begin{pmatrix}\partial_x u_\sigma \\ \partial_y u_\sigma\end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \zeta & 1 \end{pmatrix}\begin{pmatrix}\partial_x u_\sigma \\ \partial_y u_\sigma\end{pmatrix} = \begin{pmatrix}\partial_x u_\sigma \\ \zeta\,\partial_x u_\sigma + \partial_y u_\sigma\end{pmatrix} = \begin{pmatrix}\partial_x u_\sigma \\ 0\end{pmatrix},$$

which is equivalent to

$$\zeta = -\frac{\partial_y u_\sigma}{\partial_x u_\sigma} \quad \text{if} \quad \partial_x u_\sigma \neq 0.$$

See Fig. 3 for illustration. Hence, we can estimate the shear parameters by the gradients of the two non-horizontal edges of each sheared rectangle. To get only those gradients, we locate the horizontal edges and the corners of the sheared rectangles and downsize

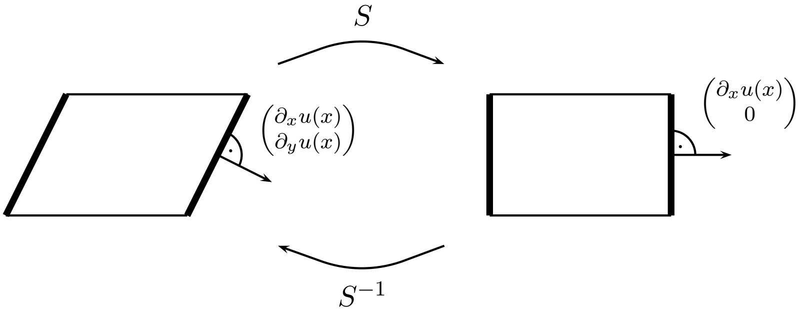
S
$\begin{pmatrix}\partial_x u(x)\\ \partial_y u(x)\end{pmatrix}$
$\begin{pmatrix}\partial_x u(x)\\ 0\end{pmatrix}$
S^{-1}

Figure 3: Performance of the shear operation on rectangles.

the gradients at those locations to reduce their influence on the structure tensor. More precisely, we compute

$$\tilde{\nabla} u_{\tilde{\sigma}} := w_e(\nabla u_{\tilde{\sigma}})\, w_v(\nabla u_{\tilde{\sigma}})\nabla u_{\tilde{\sigma}},$$

where we use the following weight functions: Let μ_1, μ_2 be the eigenvalues of the structure tensors $J_\rho(\nabla u_{\tilde{\sigma}})$. In order to *downsize* $\nabla u_{\tilde{\sigma}}$ *at vertices*, we multiply the gradient by

$$w_v(\nabla u_{\tilde{\sigma}}) := \begin{cases} \varphi_v\left(\left|\frac{\mu_2}{\mu_1}\right| |\nabla u_{\tilde{\sigma}}|^2\right) & \text{if } \mu_1 > 0, \\ 1 & \text{otherwise,} \end{cases}$$

where φ_v has the properties of a diffusivity function, i.e., it is decreasing and $\varphi_v(0) = 1$. Here, $\left|\frac{\mu_2}{\mu_1}\right| |\nabla u_{\tilde{\sigma}}|^2$ plays the role of a corner detector, since in general $\mu_1 \geq \mu_2 \gg 0$ at vertices, while $\mu_1 \gg \mu_2$ at edges, see [13, 28].

To *enforce* $\nabla u_{\tilde{\sigma}}$ *to become small at horizontal edges* we further apply

$$w_e(\nabla u_{\tilde{\sigma}}) := \begin{cases} \varphi_e(|\frac{\partial_x u_{\tilde{\sigma}}}{\partial_y u_{\tilde{\sigma}}}|) & \text{if } |\partial_y u_{\tilde{\sigma}}| > 0, \\ 1 & \text{otherwise,} \end{cases}$$

where φ_e is a sigmoidal function, i.e., it is monotone increasing with $\varphi_e(x) = 0$ for $x \leq 0$ and $\varphi_e(x) = 1$ for $x \geq 1$.

For these modified gradients we calculate the structure tensors

$$J_{(\rho_x,\rho_y)}\big(\tilde{\nabla} u_{\tilde{\sigma}}\big) := K_{(\rho_x,\rho_y)} * J_0\big(\tilde{\nabla} u_{\tilde{\sigma}}\big)$$

for a non-radial Gaussian $K_{(\rho_x,\rho_y)} := \frac{1}{2\pi\rho_x\rho_y} e^{-\left((x/\rho_x)^2+(y/\rho_y)^2\right)/2}$ and determine the eigenvectors $v = (v_1, v_2)^{\mathrm{T}}$ belonging to the largest eigenvalues. In general we propose to use an anisotropic Gaussian, where ρ_x and ρ_y are not necessarily equal to better adapt the smoothing of J_0 to the geometry. Then, we define the shear parameters ζ by

$$\zeta := \begin{cases} -\frac{v_2}{v_1} & \text{if} \quad v_1 > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Finally, we set the corresponding diffusion tensors to

$$D := \begin{pmatrix} 1 & \zeta \\ 0 & 1 \end{pmatrix} \begin{pmatrix} g(\partial_x u_\sigma) & 0 \\ 0 & g(\zeta\, \partial_x u_\sigma + \partial_y u_\sigma) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \zeta & 1 \end{pmatrix}. \tag{9}$$

Examples of computed shear parameters are depicted in Fig. 6 and 7.

3 Anisotropic regularization model

Alternatively to anisotropic diffusion we can also use regularization methods to process images consisting of linearly transformed rectangles if we know the transformation matrix A at each point $x \in \Omega$. In this case, we minimize the energy functional

$$F(u) = \frac{1}{2} \int\limits_{\Omega} (f(x) - u(x))^2 \, dx + \lambda \int\limits_{\Omega} \|A(x)\nabla u(x)\|_1 dx, \tag{10}$$

where $\lambda > 0$ is a regularization parameter.

For images containing rotated rectangles, we use the rotation matrices $A = R$ with c_ρ, s_ρ defined in Subsection 2.2, for images with sheared rectangles the shear matrices $A = S$ with shear parameters ζ determined in Subsection 2.3, and, if available, the combination $A = RS$ in case of both sheared and rotated rectangles. In contrast to our diffusion model these matrices can be computed once in advance and will not be repeatedly updated as it will be done in every diffusion step. In [4, 24], model (10) was presented for rotation matrices $A = R$, where R was constructed by minimizing certain functionals in a first step.

To see the relation to diffusion equations we consider the Euler–Lagrange equation of (10) with the slightly modified regularization term $\Phi(A\nabla u) := \sqrt{(A\nabla u)_1^2 + \varepsilon^2} + \sqrt{(A\nabla u)_2^2 + \varepsilon^2}$, $\varepsilon \ll 1$ which reads

$$\frac{u - f}{\lambda} = \operatorname{div}(D(\nabla u)\nabla u)$$

with the diffusion tensor

$$D(\nabla u) := A^{\mathrm{T}} \begin{pmatrix} 1/\sqrt{(A\nabla u)_1^2 + \varepsilon^2} & 0 \\ 0 & 1/\sqrt{(A\nabla u)_2^2 + \varepsilon^2} \end{pmatrix} A. \tag{11}$$

This can be interpreted as a fully implicit time discretization of the diffusion equation (1) but without the additional smoothing $u_\sigma = K_\sigma * u$ in the diffusion tensor. This leads to a PDE with a special TV-related diffusivity which is a 'boundary case' between forward and backward diffusion, see [9, p. 57] and [28]. In contrast, we are more flexible in (6), (8) and (9) because we can choose various diffusivities, e.g., the Perona-Malik diffusivity which leads to forward-backward diffusion. This PDE becomes well-posed by using the smoothed image u_σ in the diffusivity [7], but is ill-posed otherwise [10, 11]. For regularization effects just by discretization, see [29]. A variational formulation of the PDE with Perona-Malik diffusivity would contain a non-convex penalizing term Φ. Moreover, it is not clear how the smoothed u_σ in the diffusion tensor (and only there) can be brought into play in a variational formulation. For further relations between variational approaches and diffusion equations we refer to [23].

4 Discretization issues

Our numerical approaches are based on finite difference methods for discrete images. In the following, we briefly outline some special issues.
Let $U \in \mathbb{R}^{n \times n}$ be a discrete image and $u \in \mathbb{R}^N$ with $N = n^2$ the corresponding columnwise reshaped image vector.

Anisotropic diffusion To discretize diffusion model (1) we apply an explicit time discretization and discretize partial spatial derivatives by central differences, where we additionally use a smoothing filter of the form $\frac{1}{16}\begin{pmatrix} 3 & 10 & 3 \end{pmatrix}$ orthogonal to the derivative direction as suggested by Weickert and Scharr in [30]. As a result, we end up with an iterative scheme of the form

$$\begin{aligned} u^{(k+1)} &= u^{(k)} + \tau Q(u^{(k)}) u^{(k)}, \\ u^{(0)} &= f, \end{aligned} \tag{12}$$

where $\tau > 0$ is a fixed time step size and $Q(u^{(k)}) \in \mathbb{R}^{N \times N}$ the iteration matrix corresponding to the spatial derivatives. All non-vanishing entries of the $(i + nj)^{\text{th}}$ row of $Q(u^{(k)})$ are represented by the stencil shown in Fig. 4, where the diffusion tensor at pixel (i, j) is defined as

$$D(\nabla u_\sigma(i,j)) = \begin{pmatrix} a_{i,j} & b_{i,j} \\ b_{i,j} & c_{i,j} \end{pmatrix}.$$

An alternative to this stencil is for example the nonnegativity discretization of Weickert [28]. For a comparison see [22].

$9a_{i-1,j-1}$ $+18b_{i-1,j-1}$ $+9c_{i-1,j-1}$	$30b_{i-1,j-1} + 30b_{i-1,j}$ $+30c_{i-1,j-1} + 30c_{i-1,j}$	$-9a_{i-1,j-1} - 9a_{i-1,j+1}$ $+9c_{i-1,j-1} + 9c_{i-1,j+1}$ $+100c_{i-1,j}$	$-30b_{i-1,j} - 30b_{i-1,j+1}$ $+30c_{i-1,j} + 30c_{i-1,j+1}$	$9a_{i-1,j+1}$ $-18b_{i-1,j+1}$ $+9c_{i-1,j+1}$
$30a_{i-1,j-1} + 30a_{i,j-1}$ $+30b_{i-1,j-1} + 30b_{i,j-1}$	$100b_{i-1,j} + 100b_{i,j-1}$	$-30a_{i-1,j-1} - 30a_{i-1,j+1}$ $-30a_{i,j-1} - 30a_{i,j+1}$ $-30b_{i-1,j-1} + 30b_{i-1,j+1}$ $+30b_{i,j-1} - 30b_{i,j+1}$	$-100b_{i-1,j} - 100b_{i,j+1}$	$30a_{i-1,j+1} + 30a_{i,j+1}$ $-30b_{i-1,j+1} - 30b_{i,j+1}$
$9a_{i-1,j-1} + 9a_{i+1,j-1}$ $+100a_{i,j-1}$ $-9c_{i-1,j-1} - 9c_{i+1,j-1}$	$-30b_{i-1,j-1} + 30b_{i-1,j}$ $+30b_{i+1,j-1} - 30b_{i+1,j}$ $-30c_{i-1,j-1} - 30c_{i-1,j}$ $-30c_{i+1,j-1} - 30c_{i+1,j}$	$-9a_{i-1,j-1} - 9a_{i-1,j+1}$ $-100a_{i,j-1} - 100a_{i,j+1}$ $-9a_{i+1,j-1} - 9a_{i+1,j+1}$ $-18b_{i-1,j-1} + 18b_{i-1,j+1}$ $+18b_{i+1,j-1} - 18b_{i+1,j+1}$ $-9c_{i-1,j-1} - 9c_{i-1,j+1}$ $-100c_{i-1,j} - 100c_{i+1,j}$ $-9c_{i+1,j-1} - 9c_{i+1,j+1}$	$-30b_{i-1,j} + 30b_{i-1,j+1}$ $+30b_{i+1,j} - 30b_{i+1,j+1}$ $-30c_{i-1,j} - 30c_{i-1,j+1}$ $-30c_{i+1,j} - 30c_{i+1,j+1}$	$9a_{i-1,j+1} + 9a_{i+1,j+1}$ $+100a_{i,j+1}$ $-9c_{i-1,j+1} - 9c_{i+1,j+1}$
$30a_{i,j-1} + 30a_{i+1,j-1}$ $-30b_{i,j-1} - 30b_{i+1,j-1}$	$-100b_{i,j-1} - 100b_{i+1,j}$	$-30a_{i,j-1} - 30a_{i,j+1}$ $-30a_{i+1,j-1} - 30a_{i+1,j+1}$ $-30b_{i,j-1} + 30b_{i,j+1}$ $+30b_{i+1,j-1} - 30b_{i+1,j+1}$	$100b_{i,j+1} + 100b_{i+1,j}$	$30a_{i,j+1} + 30a_{i+1,j+1}$ $+30b_{i,j+1} + 30b_{i+1,j+1}$
$9a_{i+1,j-1}$ $-18b_{i+1,j-1}$ $+9c_{i+1,j-1}$	$-30b_{i+1,j-1} - 30b_{i+1,j}$ $+30c_{i+1,j-1} + 30c_{i+1,j}$	$-9a_{i+1,j-1} - 9a_{i+1,j+1}$ $+9c_{i+1,j-1} + 9c_{i+1,j+1}$ $+100c_{i+1,j}$	$30b_{i+1,j} + 30b_{i+1,j+1}$ $+30c_{i+1,j} + 30c_{i+1,j+1}$	$9a_{i+1,j+1}$ $+18b_{i+1,j+1}$ $+9c_{i+1,j+1}$

Figure 4: Discretization stencil (multiplied by 32^2) of $\operatorname{div}(D(\nabla u_\sigma)\nabla u)$ at pixel (i, j) as suggested in [22].

At the boundary of U we apply Neumann boundary conditions. Furthermore, we set

$$\begin{aligned} a_{i,1} = a_{i,n} = b_{i,1} = b_{i,n} = 0 \quad \forall i \in \{1, \cdots n\} \\ b_{1,j} = b_{n,j} = c_{1,j} = c_{n,j} = 0 \quad \forall j \in \{1, \cdots n\} \end{aligned}$$

to guarantee that $\left(D(\nabla u_\sigma(i,j)) \nabla u(i,j) \right)^{\mathrm{T}} n(i,j) = 0$ is fulfilled at the boundaries. The sum over all entries of the stencil shown in Fig. 4 is zero, since the central entry is the negative sum of the other ones. Combined with our Neumann boundary conditions, this guarantees that the sum of all entries in one row of $Q(u^{(k)})$ is zero. Since $Q(u^{(k)})$ is symmetric, it follows that the sum of all elements of one column of $Q(u^{(k)})$ is also zero. Hence, the iterative scheme (12) preserves the average gray value, i.e.,

$$\frac{1}{N} \sum_{i=1}^{N} f_i = \frac{1}{N} \sum_{i=1}^{N} u_i^{(k)}, \quad k \in \mathbb{N}.$$

However, since there may appear negative matrix entries in $I + \tau Q(u^{(k)})$, the iterates do not in general fulfill a min-max principle.

To avoid possible checkerboard effects which may appear in the presence of heavy noise, the scheme (12) can be extended by the additional term $\nu(\tilde{I} - I)u^{(k)}$ with a very small parameter ν. The low pass filter matrix $\tilde{I}$ 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 of consistency order 4, i.e., it has no influence on the consistency order of the original scheme. For more details see [22, Section 9.5]. This additive term leads again to a symmetric iteration matrix with a row sum of zero. Hence, the resulting scheme preserves the average gray value, too.

Anisotropic regularization To minimize (10) numerically, we compute the minimizer of its discrete counterpart

$$\frac{1}{2} \|f - u\|_2^2 + \lambda \| \underbrace{\mathcal{A} D}_{M} u \|_1, \tag{13}$$

where $f, u \in \mathbb{R}^N$ are image vectors, $D \in \mathbb{R}^{2N,N}$ is a discrete partial derivative operator and $\mathcal{A} \in \mathbb{R}^{2N,2N}$ a matrix containing the transformation matrices $A(i,j)$. In our case we set D to be the derivative filters defined in [24, Section 2.1]. Moreover, to avoid possible checkerboard effects, we extend (13) by the additional term $\nu \|(H_1 \otimes H_1) u\|_1$, where ν is a very small parameter and H_1 is the upper 2-band Toeplitz matrix with band $(-1, 1)$. This additional term penalizes diagonal differences. For more details see [24, Section 2.2]. As a consequence we obtain a slightly modified matrix M.

The minimizer $\hat{u}$ of (13) can be alternatively characterized by the dual formulation $\hat{u} = f - M^{\mathrm{T}} \hat{V}$, where $\hat{V} \in \mathbb{R}^{2N}$ is a minimizer of the quadratic functional with linear constraints

$$\min_{V \in \mathbb{R}^{2N}} \|f - M^{\mathrm{T}} V\|_2^2 \qquad \text{subject to} \qquad \|V\|_\infty \le \lambda. \tag{14}$$

There exist various numerical methods to solve (14). In this paper, we prefer, due to the observed fast convergence, to minimize the functional (13), resp. (14) by second-order cone programming (SOCP). Concerning SOCP we refer to [3, 17]. To determine a solution $\hat{V}$ of (14) by SOCP we reformulate the problem appropriately as

$$\min_{t\in\mathbb{R}, V\in\mathbb{R}^{2N}} \begin{pmatrix} t \\ \\ V \\ \\ \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \text{subject to} \quad \begin{pmatrix} f - M^{\mathrm{T}}V \\ t \\ 1 \end{pmatrix} \in L_r^{N+2},$$

$$-V_i, V_i \leq \lambda, \quad i \in \{1, \cdots, 2N\}$$

where L_r^{N+2} denotes the rotated second-order cone defined by

$$L_r^{N+2} := \left\{ (\bar{x}, x_{N+1}, x_{N+2})^{\mathrm{T}} \in \mathbb{R}^{N+2} : \|\bar{x}\|_2^2 \leq 2x_{N+1}x_{N+2},\ x_{N+2} \geq 0 \right\}.$$

5 Numerical examples

In the following, we present numerical examples. The implementation was done in MATLAB, where we additionally used the software package MOSEK [1] for SOCP. To visualize the images we have applied the MATLAB routine 'imagesc' which incorporates an affine gray value scaling. Besides, the parameters were chosen with respect to the best visual results. In all diffusion experiments, if not stated otherwise, we have used the weight $\nu = 0.001$ to avoid checkerboard effects, time step size $\tau = 0.1$ and 4000 iterations , i.e., a diffusion time of $t = 400$.

In our first experiment, we denoised the image at top middle of Fig. 1 with anisotropic diffusion using the adapted diffusion tensor (6) for rotated rectangles. The result is shown at top right of Fig. 5. In contrast to the denoising results presented in Fig. 1 the vertices of the rectangles are well preserved. As illustrated at Fig. 5 bottom left, this is also the case if we apply the anisotropic regularization method presented in Section 3. For comparison we also applied two times the iterated NL means from the software package [16] with parameters $t = 30$, $f = 10$ and $h = 4000$. Since this method uses the information of similar patches in the image it shows also a good performance in the presence of heavy noise.

For our next examples, we used the results of Section 2.3 to denoise arbitrary sheared rectangles. For φ_v, φ_e we chose the functions

$$\begin{aligned} \varphi_v(x) &:= e^{-(x/c)^4}, \qquad c := 0.06 \max_{x\in\Omega}\{\left|\frac{\mu_2(x)}{\mu_1(x)}\right| |\nabla u_\sigma|^2(x)\}, \quad \mu_1 > 0, \\ \varphi_e(x) &:= \sin\left(\frac{\pi}{2}x\right), \qquad x \in [0,1] \end{aligned}$$

with μ_1, μ_2 defined in Section 2.3. As shown at top right of Fig. 6 the shear parameters are well estimated from the noisy image at top left. Moreover, our methods preserve the vertices of the parallelograms as depicted at the bottom of Fig. 6.

As demonstrated by Fig. 7 our method can also be applied for more general shapes. Here, we have estimated the shear parameters of an image which contains no transformed rectangles, but all vertices are sheared rectangular ones. Then we have incorporated these

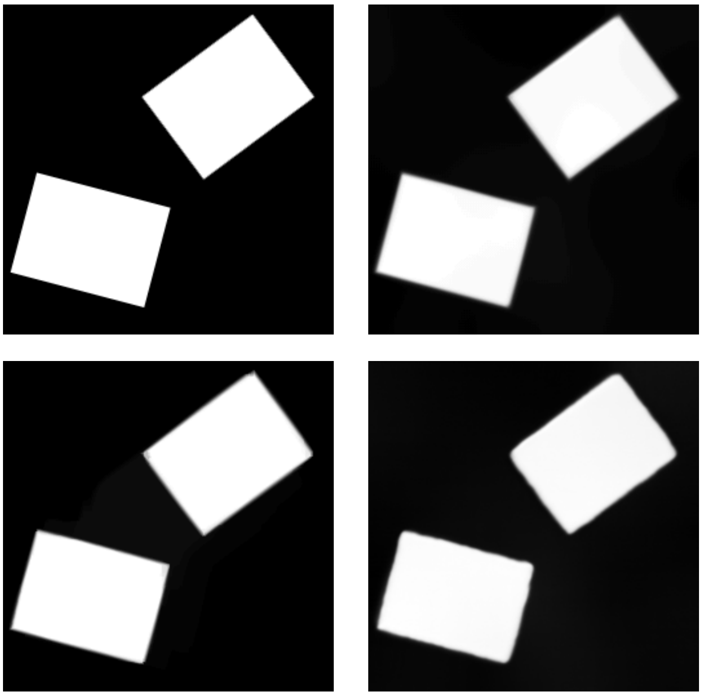

Figure 5: Denoising results for the rectangles at top middle of Fig. 1. Top left: original image. Top right: denoised image by anisotropic diffusion with adapted diffusion tensors (6) for rotated rectangles ($\sigma = 2.5$, $\rho = 20$) and Perona-Malik diffusivity ($\gamma = 1.5$). Bottom left: denoised image using the regularization method proposed in Section 3 with $\sigma = 2.5$, $\rho = 20$, $\lambda = 800$ and $\nu = 0.1$. Bottom right: denoised image using two times iterated NL means. The result is slightly worse than those of our methods.

estimates to restore the noisy image which leads to very good results. To test our method also on real–world data we have used the image depicted in Fig. 8. The results show that the proposed method performs much better than ROF for this example, since by the estimated shear parameters the shapes of the shadows are preserved much more accurate.

Regarding possible applications we have applied our new angle estimation method to an image with differently oriented toy cars. As visible at Fig. 9 right, the orientations of the cars are well estimated by our new approach and through the color coding of the angles equally oriented cars are grouped together. In connection with quality control this might for example be helpful to detect wrongly oriented objects very easily.

Another application of our methods for rotated rectangular shapes is presented in our last example. Here, we have extracted the cartoon of a real–world image of a city area with the anisotropic regularization method of Section 3. The result presented in Fig. 10 top right shows that the estimated rotation angles give a good approximation of the orientations of the houses. For this reason, the shapes of the houses are well preserved in the cartoons while the details are removed. If we now apply the Canny edge detector [6] implemented

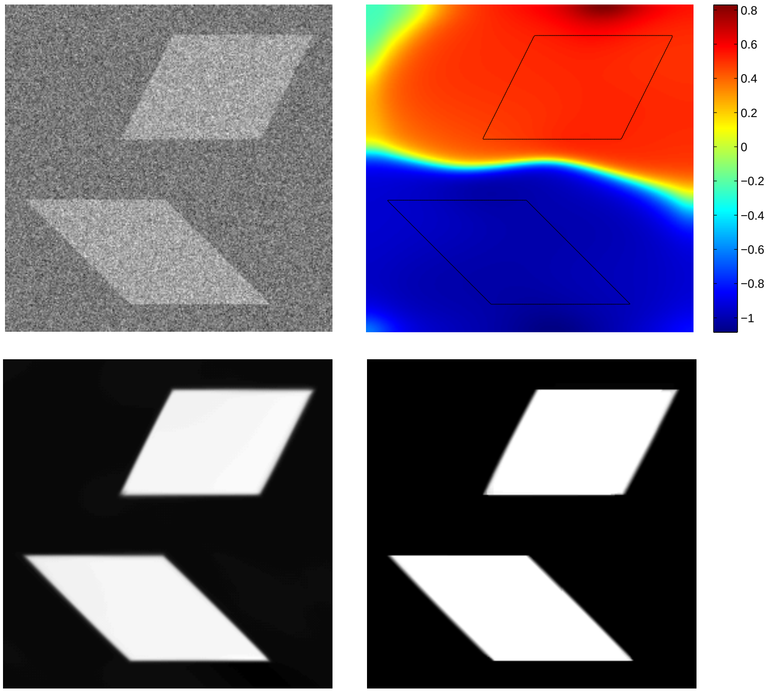

Figure 6: Top left: noisy image of two sheared rectangles ($\zeta_1 = 0.5$ and $\zeta_2 = -1$) corrupted by white Gaussian noise of standard deviation 150. Top right: estimated shear parameters ζ for $\tilde{\sigma} = 5$, $\rho = 2$, $\rho_x = 35$ and $\rho_y = 20$. Bottom left: denoised image by anisotropic diffusion with adapted diffusion tensors (9) to sheared rectangles ($\sigma = 2.5$) and Perona-Malik diffusivity ($\gamma = 1.5$). Bottom right: denoised image using the regularization method proposed in Section 3 with shear parameters ζ as depicted at top right, $\lambda = 1000$ and $\nu = 0.1$.

in the MATLAB 'edge' routine to the original image as well as to our simplified one, we see that the shapes of the houses are well extracted from the simplified image without the details contained in the original image. The obtained edge image can now be used for e.g. building segmentation. A first overview of the vast literature on this topic can be found in [15]. To give an example we have implemented the windowed Hough transform for extracting rectangles presented in [8]. The results are depicted in Fig. 11 and 12. By using the edge image of the cartoon in Fig. 10 bottom middle instead of the original image we benefit from a reduction in the computational time needed for the segmentation as well as a lower risk of detecting false positives.

6 Summary and Conclusions

Preserving vertices is still a problem in image processing. We have presented anisotropic diffusion methods for processing images containing rotated or sheared rectangles which

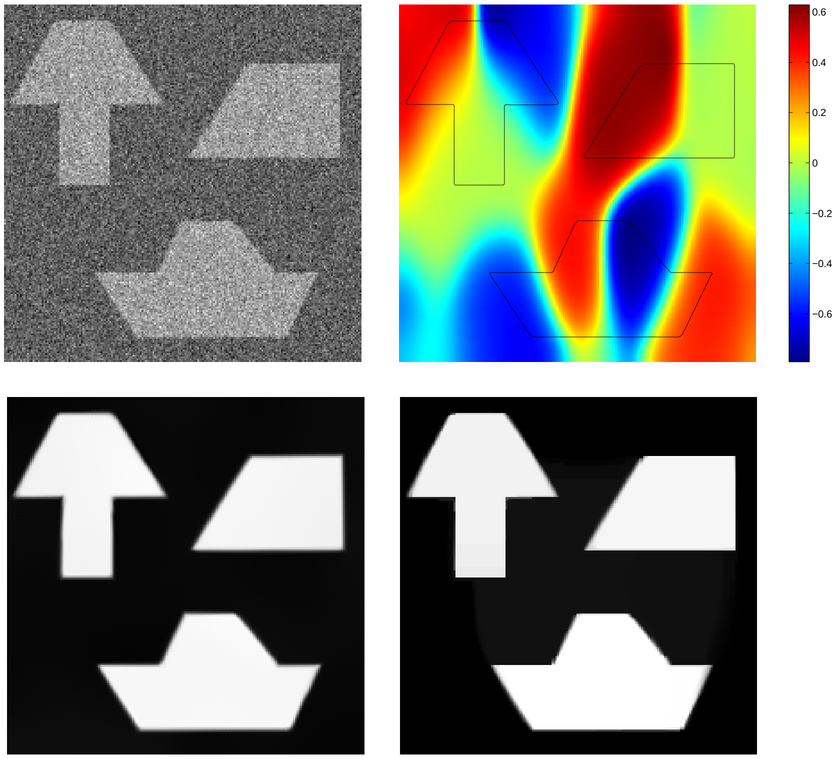

Figure 7: Top left: noisy image corrupted by white Gaussian noise of standard deviation 100. Top right: estimated shear parameters ζ for $\tilde{\sigma} = 2$, $\rho = 1$, $\rho_x = 25$ and $\rho_y = 12$. Bottom left: denoised image by anisotropic diffusion after 3000 iterations ($\sigma = 2$) with adapted diffusion tensors (9) to sheared rectangles and Perona-Malik diffusivity ($\gamma = 1.5$). Bottom right: denoised image using the regularization method proposed in Section 3 with the shear parameters at top right, $\lambda = 600$ and $\nu = 0.07$.

solve this problem using special diffusion tensors. These diffusion tensors are derived by different adaptations of the structure tensor guiding the diffusion at vertices in the directions of the corresponding edges. Moreover, the modified structure tensors can be used for anisotropic regularization as outlined in Section 3.
Future work has to be invested for processing images containing both rotated and sheared rectangles as well as arbitrary multiple orientations at corners and junctions. Here, structure tensors containing for example higher-order derivatives maybe useful [2].

References

[1] MOSEK Optimization Toolbox. `http://www.mosek.com`.

[2] T. Aach, C. Mota, I. Stuke, M. Mühlich, and E. Barth. Analysis of superimposed oriented patterns. *IEEE Transactions on Image Processing*, 15(12):3690–3700, 2006.

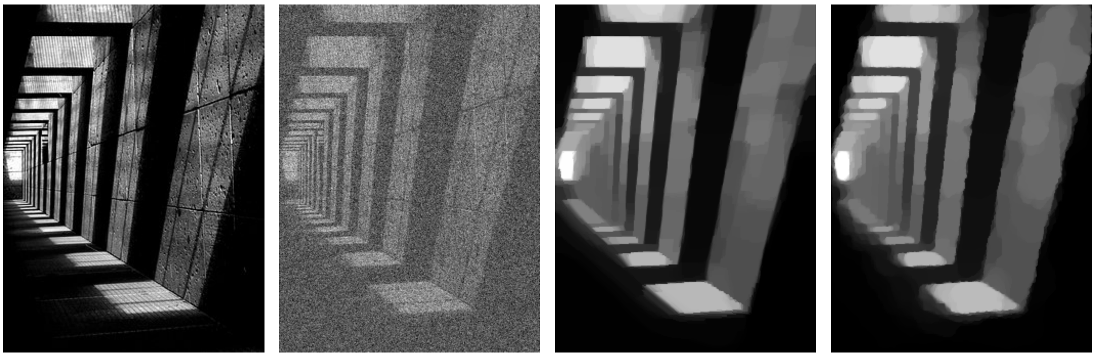

Figure 8: Left to right: original image (copyright by [14]), noisy image corrupted by white Gaussian noise of standard deviation 70, a restored image by the anisotropic regularization method proposed in Section 3 with estimated shear parameters ($\tilde{\sigma} = 6$, $\rho = 1$, $\rho_x = \rho_y = 20$, $\lambda = 150$, $\nu = 0.1$), and for comparsion a denoised image by ROF with $\lambda = 200$.

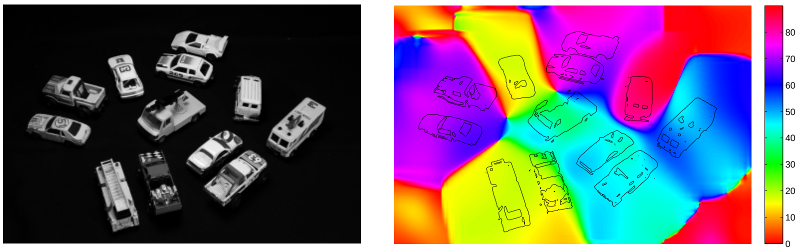

Figure 9: Left: real image of differently oriented toy cars. Right: estimated orientations by our new structure tensor method for rotated shapes with parameters $\sigma = 0.5$ and $\rho = 15$.

[3] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical Programming*, 95:3–51, 2003.

[4] B. Berkels, M. Burger, M. Droske, O. Nemitz, and M. Rumpf. Cartoon extraction based on anisotropic image classification. In *Vision, Modeling, and Visualization Proceedings*, pages 293–300. 2006.

[5] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *IEEE Int. Conf. on Computer Vision and Pattern Recignition, CVPR*, volume 2, pages 60–65, 2005.

[6] F. J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[7] 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.

[8] D. Cireşan and D. Damian. Preserving topological information in the windowed Hough transform for rectangle extraction. In K. Franke et al., editor, *Pattern Recognition, 28th DAGM Symposium*, volume 4174 of *LNCS*, pages 172–181. Springer, 2006.

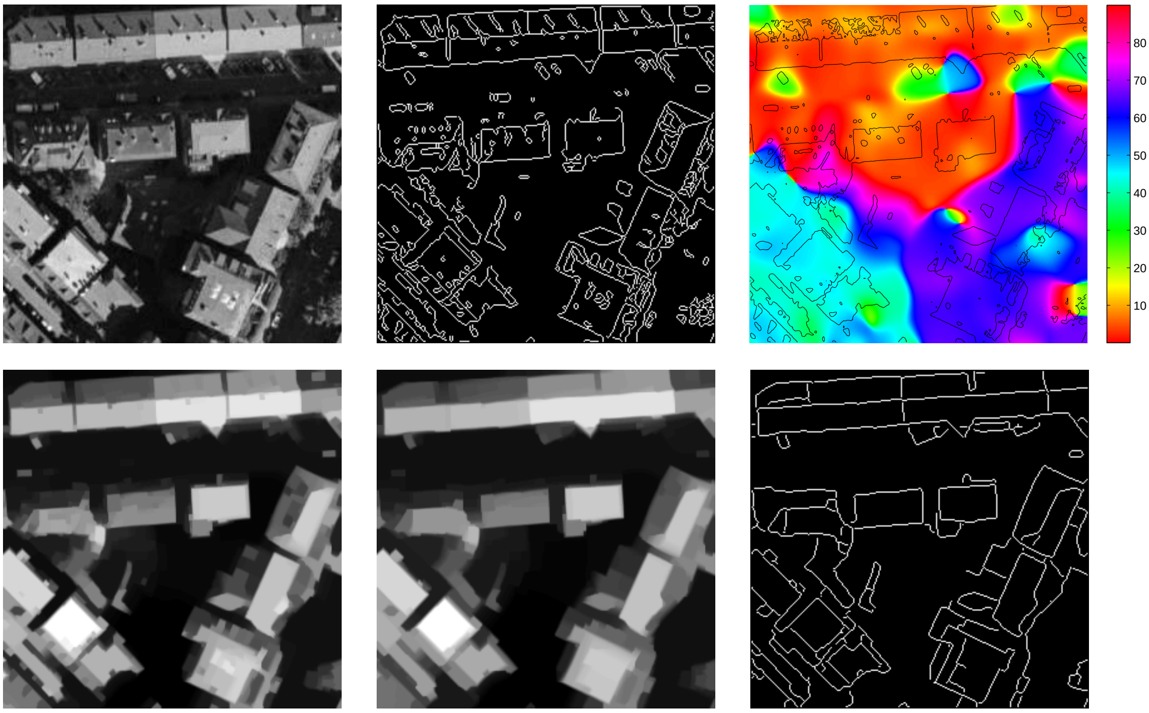

Figure 10: Top left: original image of a city area of Stuttgart. Top middle: edges detected by the Canny edge detector in the original image ($\sigma = 0.5$, thresh $= [0.1,\ 0.3]$). Top right: estimated rotation angles by the adapted structure tensor (6) for rotated rectangles ($\sigma = 0.5$, $\rho = 10$). Bottom left/middle: cartoons generated by anisotropic regularization using the rotation angles at top right, $\nu = 0.1$ and parameters $\lambda = 50$ (left) and $\lambda = 100$ (middle). Bottom right: edges detected by the Canny edge detector in the cartoon at bottom middle ($\sigma = 0.5$, thresh $= [0.03,\ 0.12]$).

[9] S. Didas. *Denoising and Enhancement of Digital Images.* Ph.D. Thesis, Saarland University, Saarbrücken, Germany, 2007.

[10] S. Esedoglu. An analysis of the Perona-Malik scheme. *Communications in Pure and Applied Mathematics*, 54:1442–1487, 2004.

[11] S. Esedoglu. Stability properties of the Perona-Malik scheme. *SIAM Journal on Numerical Analysis*, 44(3):1297–1313, 2006.

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

[13] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305, Interlaken, Switzerland, June 1987.

[14] M. Goldbach. Image "Durchgang". `http://www.photocase.com/de/upload/02/rdva4ha8/photocasefknsx7u23.jpg`.

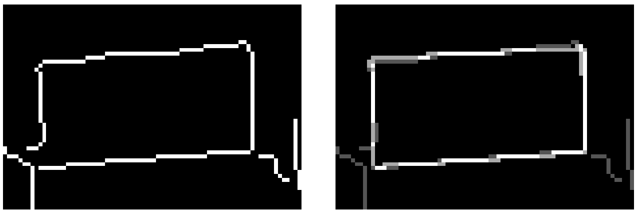

Figure 11: Left: part of the detected edges at Fig. 10 bottom right. Right: detected edges + segmented building (white line) by the windowed Hough transform presented in [8].

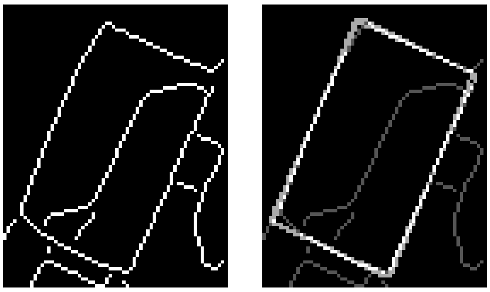

Figure 12: Left: another part of the detected edges at Fig. 10 bottom right. Right: detected edges + segmented building (white line) by the windowed Hough transform presented in [8].

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

[16] J. V. Manjón and A. Buades. MATLAB software: NL means. `http://dmi.uib.es/~abuades/software.html`.

[17] H. Mittelmann. An independent bechmarking of SDP and SOCP solvers. *Mathematical Programming Series B*, 95(2):407–430, 2003.

[18] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. *MIT Technical Report*, MIT-CSAIL-TR-2006-073, 2006.

[19] 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.

[20] G. Plonka-Hoch and J. Ma. Nonlinear regularized reaction-diffusion filters for denoising of images with textures. *IEEE Transactions on Image Processing*, 17(8):1283–1294, 2008.

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

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

[23] O. Scherzer and J. Weickert. Relations between regularization and diffusion filtering. *Journal of Mathematical Imaging and Vision*, 12(1):43–63, 2000.

[24] S. Setzer, G. Steidl, and T. Teuber. Restoration of images with rotated shapes. *Numerical Algorithms*, 48(1):49–66, 2008.

[25] A. Spira, R. Kimmel, and N. Sochen. A short-time Beltrami kernel for smoothing images and manifolds. *IEEE Transactions on Image Processing*, 16(6):1628–1636, 2007.

[26] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proc. Sixth International Conference on Computer Vision*, pages 839–846, Bombay, India, Jan. 1998. Narosa Publishing House.

[27] D. Tschumperlé. Fast anisotropic smoothing of multivalued images using curvature preserving PDEs. *International Journal of Computer Vision*, 68(1):65–82, 2006.

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

[29] J. Weickert and B. Benhamouda. A semidiscrete nonlinear scale-space theory and its relation to the Perona–Malik paradox. In F. Solina, W. G. Kropatsch, R. Klette, and R. Bajcsy, editors, *Advances in Computer Vision*, pages 1–10. Springer, Wien, 1997.

[30] 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.