Restoration of Matrix Fields by Second Order Cone Programming

G. Steidl S. Setzer B. Popilka* B. Burgeth†

February 1, 2007

Abstract

Wherever anisotropic behaviour in physical measurements or models is encountered matrices provide adequate means to describe this anisotropy. Prominent examples are the diffusion tensor magnetic resonance imaging in medical imaging or the stress tensor in civil engineering. As most measured data these matrix-valued data are also polluted by noise and require restoration.
The restoration of scalar images corrupted by noise via minimization of an energy functional is a well-established technique that offers many advantages. A convenient way to achieve this minimization is second order cone programming (SOCP). The goal of this article is to transfer this method to the matrix-valued setting. It is shown how SOCP can be applied to minimize various energy functionals defined for matrix fields. These functionals couple the different matrix channels taking into account the relations between them. Furthermore, new functionals for the regularization of matrix data are proposed and the corresponding Euler-Lagrange equations are derived by means of matrix differential calculus. Numerical experiments substantiate the usefulness of the proposed methods for the restoration of matrix fields.

1 Introduction

Matrix-valued data, so-called matrix fields have gained significant importance in recent years:

- First, *diffusion tensor magnetic resonance imaging (DT-MRI)* [3] is a modern but commonly used medical imaging technique that measures a 3×3 positive semidefinite matrix-field: A so-called diffusion tensor is assigned to each voxel. This diffusion tensor describes the diffusive property of water molecules. Since water diffuses preferably along ordered tissue such as nerve fibers this matrix gives valuable information

*University of Mannheim, Dept. of Matematics and Computer Science
†Saarland University, Mathematical Image Analysis Group

about the geometry and organization of the tissue under examination. Hence this matrix field plays a very important role for the diagnosis of multiple sclerosis and strokes. For detailed information about the acquisition of this type of data the reader is referred to [2] and the literature cited therein.

- Second, in the field of technical sciences such as civil engineering and solid mechanics or geology anisotropic behaviour is often described satisfactorily by inertia, diffusion, stress, and permittivity tensors.

- Third, matrices/tensors have been recognized as a useful concept in image analysis itself [15]: The *structure tensor* [13], for instance, (also called Förstner interest operator, or scatter matrix) has been employed not only for corner detection [16], but also for texture analysis [23] and motion estimation [5]. Tensor voting, an interesting recent tool for segmentation and grouping, also makes use of the tensor concept.

This variety of applications make it worthwhile to develop appropriate tools for the restoration and processing of tensor, respectively matrix data, since, just as scalar images, they are corrupted by noise. However, when designing filters for matrix fields, treating the channels independently is a simple though not advisable strategy. Any relation between the different matrix channels is ignored which leads to similarly serious shortcomings as in the case of vector-valued filtering.

Unlike vectors, matrices can be multiplied making matrix-valued polynomials and also functions of matrices. These useful notions that rely decisively on the strong interplay between the different matrix entries. Roughly speaking, we are taking an operator-algebraic point of view here concentrating on symmetric matrices as finite-dimensional instances of self-adjoint operators. Unfortunately in the case of those symmetric matrices, extra care has to be taken since the product of two symmetric matrices is usually not symmetric: The Jordan product is used as an symmetric multiplication. In fact this product makes its natural appearance in the derivation of energy functionals used in this paper for matrix field restoration.

This paper is organized as follows: Since we want to convert restoration methods which were successfully applied in the scalar valued case to the matrix–valued setting, we start by considering the related scalar–valued techniques in Section 2. Section 3 provides preliminaries on matrix-valued functions and introduces to second order cone programming (SOCP). In Section 4 we examine properties of a functional suggested by Deriche and Tschumperlé for the root function in the penalizing term and show how SOCP can be applied to find the minimizer of this functional. Section 5 proposes two new functionals which better correspond to the matrix structure of our objects. The corresponding Euler-Lagrange equation includes the Jordan product of matrices. We apply SOCP and a steepest decent method

to compute minimizers of these functionals. Finally, Section 6 compares the different methods by various numerical examples.

2 Motivation: restoration of scalar-valued functions

A well-established method for restoring a scalar-valued image u from a given degraded image f consists in calculating the minimizer of the functional

$$\mathcal{J}(u) := \frac{1}{2} \int_\Omega (f-u)^2 + \alpha\, \Phi(|\nabla u|^2)\, dxdy \tag{1}$$

with regularization parameter $\alpha > 0$ and an increasing function $\Phi : [0,\infty] \to \mathbb{R}$ in the penalizing term. The first summand encourages similarity between the restored image and the original one, while the second one rewards smoothness. The appropriate choice of the function Φ ensures that important image structures such as edges are preserved while areas with small gradients are smoothed. A standard way for solving (1) uses the fact that the minimizer has to fulfill the Euler–Lagrange equation

$$0 = f - u + \alpha \operatorname{div}(\Phi'(|\nabla u|^2)\nabla u).$$

Then a steepest descent method can be applied which is equivalent to computing the steady state of the reaction-diffusion equation

$$\partial_t u = f - u + \alpha \operatorname{div}(\Phi'(|\nabla u|^2)\nabla u)$$

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

$$\frac{u-f}{\alpha} = \operatorname{div}(\Phi'(|\nabla u|^2)\nabla u).$$

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

$$\partial_t u = \operatorname{div}(\Phi'(|\nabla u|^2)\nabla u) \tag{2}$$

with initial image $u(\cdot,0) = f$ and homogeneous Neumann boundary conditions. The solution of this diffusion equation is a good approximation of the minimizer of (1). For details see [25, 29].

The steepest descent method requires that the function Φ is differentiable. In this paper, we are interested in the function

$$\Phi(s^2) := |s| \tag{3}$$

which is not differentiable at zero. Then the convex functional (1) is the frequently applied ROF–model introduced by Rudin, Osher and Fatemi [24].

If we want to apply a steepest descent method we have to introduce a small additional parameter ε and to deal with

$$\Phi(s^2) = \sqrt{s^2 + \varepsilon^2}, \tag{4}$$

instead of the original function, cf. [28]. However, for the function (3), the penalizing functional in (1) has very useful properties, in particular it is positive homogeneous. Based on these properties various numerical methods can be applied to find the minimizer without introducing the additional parameter ε, e.g.,

- second order cone programming (SOCP) [14],
- Chambolle's descent algorithms for the dual functional [10],
- a four pixel method for the corresponding diffusion equation [26].

In this paper, we will see how in particular SOCP can also be applied to tensor-valued images.

In addition to the functional (1), the functional

$$\mathcal{J}(u) := \frac{1}{2} \int_\Omega (f-u)^2 + \alpha \left(\Phi(u_x^2) + \Phi(u_y^2)\right) dxdy \tag{5}$$

was applied for image restoration also with higher order derivatives in literature [11, 17, 19]. For the absolute value function Φ, this functional can be handled more efficiently than (1). However, it is not rotationally invariant but may be useful for images whose edges are straight lines in connection with other techniques [4].

3 Preliminaries

Matrix-valued functions. To deal with matrix fields we have to introduce some notation. Let $\mathrm{Sym}_n(\mathbb{R})$ be the vector space of symmetric $n \times n$ matrices. This can be treated as a Euclidian vector space relative to the trace inner product

$$\langle A, B \rangle := \operatorname{tr} AB.$$

Then

$$\langle A, A \rangle = \operatorname{tr} A^2 = \|A\|_F^2$$

is the squared Frobenius norm of A. In $\mathrm{Sym}_n(\mathbb{R})$, the positive semi-definite matrices $\mathrm{Sym}_n^+(\mathbb{R})$ form a closed convex set whose interior consists of the positive definite matrices. More precisely, $\mathrm{Sym}_n^+(\mathbb{R})$ is a cone with base $\mathcal{B}$ [1, 8, 9], i.e.

$$\mathrm{Sym}_n^+(\mathbb{R}) = \mathbb{R}_{\geq 0}\, \mathcal{B}$$

and

$$\mathcal{B} := \{B \in \mathrm{Sym}_n^+(\mathbb{R}) : \mathrm{tr}\, B = 1\}.$$

Since $\mathcal{B}$ is a convex compact set in a finite dimensional space it is, by the Krein-Milman theorem, the convex hull of its extreme points which are given by the rank 1 matrices vv^{T} with $\|v\|_2 = 1$. Thus,

$$\mathcal{B} = \mathrm{convexhull}\{vv^{\mathrm{T}} : v \in S^{n-1}\}.$$

For $n = 2$, this can be illustrated as follows: we embed $\mathrm{Sym}_2(\mathbb{R})$ into $\mathbb{R}^3$ by

$$A \mapsto a := \frac{1}{\sqrt{2}}(2a_{12}, a_{11} - a_{22}, a_{11} + a_{22})^{\mathrm{T}}. \tag{6}$$

This mapping is an isometry from $\mathrm{Sym}_2(\mathbb{R})$ with the Frobenius norm onto $\mathbb{R}^3$ with the Euclidian norm. For $A \in \mathrm{Sym}_2^+(\mathbb{R})$ with eigenvalues $\lambda_1, \lambda_2 \geq 0$, we have that

$$\begin{aligned} \lambda_1 + \lambda_2 &= \mathrm{tr}A = a_{11} + a_{22} = \sqrt{2}\, a_3 \geq 0, \\ \lambda_1 \lambda_2 &= \det A = \frac{1}{4}\big((a_{11} + a_{22})^2 - (a_{11} - a_{22})^2 - 4a_{12}^2\big) \geq 0. \end{aligned}$$

Hence, $A \in \mathrm{Sym}_2^+(\mathbb{R})$ if and only if $a_3 \geq 0$ and $\|(a_1, a_2)^{\mathrm{T}}\|_2 \leq a_3$, i.e., the symmetric positive semi-definite matrices form the cone $\mathcal{C}^3 := \{a \in \mathbb{R}^3 : \|(a_1, a_2)^{\mathrm{T}}\|_2 \leq a_3\}$ depicted in Fig. 1. Its base $\mathcal{B}$ is just the closed disc at the height $1/\sqrt{2}$ and the extreme points form the boundary of this disc. For our numerical examples we will further use that the positive definite matrices $A \in \mathrm{Sym}_2^+(\mathbb{R})$ can be visualized as ellipses

$$\{x \in \mathbb{R}^2 : x^{\mathrm{T}} A^{-2} x = 1\}$$

whose axes have just the length of the eigenvalues of A.

By $A \circ B$ we denote the *Hadamard product* (componentwise product) and by $A \otimes B$ the *Kronecker product* (tensor product) of A and B, cf. [18]. Further we consider the so-called *Jordan–product* of matrices $A, B \in \mathrm{Sym}_n(\mathbb{R})$ defined by

$$A \bullet B := \frac{1}{2}(AB + BA) \quad \in \mathrm{Sym}_n(\mathbb{R}).$$

In contrast to the ordinary matrix multiplication the Jordan–product preserves the symmetry of the matrices. This does not hold for positive semi-definiteness. Finally, we set

$$\mathrm{vec}A := \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}$$

for an $n \times n$ matrix A with j-th column a_j.

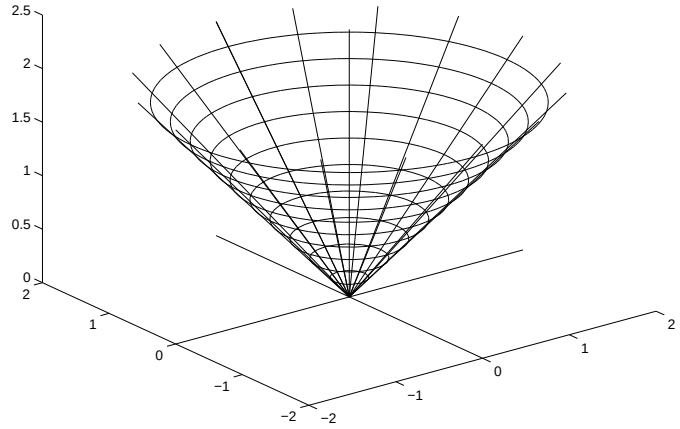

Figure 1: Cone of symmetric, positive semi-definite matrices via (6).

Second order cone programming. SOCP [20] amounts to minimize a linear objective function subject to the constraints that several affine functions of the variables have to lie in a *second order cone* $\mathcal{C}^{n+1} \subset \mathbb{R}^{n+1}$ defined by the convex set

$$\mathcal{C}^{n+1} = \left\{ \begin{pmatrix} x \\ \bar{x}_{n+1} \end{pmatrix} = (x_1, \ldots, x_n, \bar{x}_{n+1})^{\mathrm{T}} : \|x\|_2 \leq \bar{x}_{n+1} \right\}. \tag{7}$$

With this notation, the general form of a SOCP is given by

$$\inf_{x \in \mathbb{R}^n} \; f^{\mathrm{T}} x \,, \text{ s.t. } \begin{pmatrix} A_i x + b_i \\ c_i^{\mathrm{T}} x + d_i \end{pmatrix} \in \mathcal{C}^{n+1} \,, \; i = 1, \ldots, r. \tag{8}$$

Alternatively, one can also use the rotated version of the standard cone:

$$\mathcal{K}^{n+2} := \left\{ \left(x, \bar{x}_{n+1}, \bar{x}_{n+2}\right)^{\mathrm{T}} \in \mathbb{R}^{n+2} : \|x\|_2^2 \leq 2\, \bar{x}_{n+1} \bar{x}_{n+2} \right\}.$$

This allows to incorporate quadratic constraints. Problem (8) is a convex program for which efficient, large scale solvers are available [22].

4 SOCP for the Deriche-Tschumperlé functional

Let $F : \mathbb{R}^2 \to \mathrm{Sym}_n^+(\mathbb{R})$ be a matrix field corrupted by white Gaussian noise. In analogy to (1), Deriche and Tschumperlé [27] proposed to find the restored function U by minimizing the functional

$$\mathcal{J}(U) := \int_\Omega \|F - U\|_F^2 \, dxdy \; + \; \alpha \, J(U), \tag{9}$$

where

$$J(U) := \int_\Omega \Phi\big(\mathrm{tr}(U_x^2 + U_y^2)\big)\, dxdy = \int_\Omega \Phi\big(\sum_{j,k=1}^{n} \nabla u_{jk}^{\mathrm{T}} \nabla u_{jk}\big)\, dxdy. \qquad (10)$$

The penalizing term $J(U)$ contains a coupling between the matrix coefficients.

For differentiable Φ, the corresponding Euler–Lagrange equation reads

$$0 = F - U + \alpha \left(\partial_x(\Phi'(\mathrm{tr}(U_x^2 + U_y^2))U_x + \partial_y(\Phi'(\mathrm{tr}(U_x^2 + U_y^2))U_y\right).$$

In [7] this system was considered similarly as in the scalar-valued case (2) as explicit time discretization of an isotropic matrix–valued diffusion process. Based on the extremum principle fulfilled by the solution of this PDE the authors showed that the solution of the matrix–valued equation preserves for appropriate Φ the positive definiteness of the initial matrix field.

In this paper, we restrict our attention to the absolute value function Φ in (3) and SOCP. For computations, we consider the discrete counterpart of (9), where we replace the derivative operators by simple forward difference operators

$$\begin{aligned} \mathcal{J}_d(U) &:= \sum_{i,j=0}^{N-1} \|F(i,j) - U(i,j)\|_F^2 \;+\; \alpha\, J_d(U), \qquad (11)\\ J_d(U) &:= \sum_{i,j=0}^{N-1} \left(\|U(i,j) - U(i-1,j)\|_F^2 + \|U(i,j) - U(i-1,j)\|_F^2\right)^{1/2} \end{aligned}$$

with $U(-1,j) = U(i,-1) = 0$. Other discretizations of the first order derivatives are possible, too. The functional (11) is strictly convex so that it has a unique minimizer.

For sufficiently large α, we see that $\hat{U}$ minimizes $\mathcal{J}_d$ iff $J_d(\hat{U}) = 0$, i.e. $\hat{U}(i,j) = \hat{U}(0,0)$ for all $i,j = 0,\dots,N-1$. Then the data fitting term becomes minimal iff

$$\hat{U}(0,0) = \frac{1}{N^2} \sum_{j,k=0}^{N-1} F(i,j). \qquad (12)$$

We say that the discrete matrix field $F : \mathbb{Z}_N^2 \to \mathrm{Sym}_n^+(\mathbb{R})$ has all eigenvalues in an interval $\mathcal{I}$ if every matrix $F(i,j)$ of the field has all eigenvalues in $\mathcal{I}$. By the following proposition the minimizer of (11) preserves positive definiteness.

Proposition 4.1. *Let all eigenvalues of $F : \mathbb{Z}_N^2 \to \mathrm{Sym}_n^+(\mathbb{R})$ be contained in the interval $[\lambda_{\min}, \lambda_{\max}]$. Then the minimizer $\hat{U}$ of (11) has all eigenvalues in $[\lambda_{\min}, \lambda_{\max}]$.*

Proof. Using that the minimal and maximal eigenvalues $\lambda_{\min}(A), \lambda_{\max}(A)$ of a symmetric matrix A fulfill

$$\lambda_{\min}(A) = \min_{\|v\|=1} v^{\mathrm{T}} A\, v, \quad \lambda_{\max}(A) = \max_{\|v\|=1} v^{\mathrm{T}} A\, v,$$

it is easy to check that the set $\mathcal{C}$ of matrices having all eigenvalues in $[\lambda_{\min}, \lambda_{\max}]$ is convex and closed.

Assume that some matrices $\hat{U}(i,j)$ are not contained in $\mathcal{C}$. Let $P\hat{U}(i,j)$ denote the orthogonal projection (w.r.t. the Frobenius norm) of $\hat{U}(i,j)$ onto $\mathcal{C}$. Then we obtain by the projection theorem [12, p. 269] that

$$\begin{aligned} \|F(i,j) - P\hat{U}(i,j)\|_F &\leq \|F(i,j) - \hat{U}(i,j)\|_F, \\ \|P\hat{U}(i,j) - P\hat{U}(k,l)\|_F &\leq \|\hat{U}(i,j) - \hat{U}(k,l)\|_F. \end{aligned}$$

Consequently, $\mathcal{J}_d(P\hat{U}) \leq \mathcal{J}_d(\hat{U})$ which contradicts our assumption since the minimizer is unique. This completes the proof. □

Remark 4.2. *To get an intuition, let us just compute the minimizer of* (11) *for two given matrices* $F(i)$, $i = 0, 1$ *in one dimension. Via the embedding* (6) *we obtain* $F(i) \mapsto f := (f_1(i), f_2(i), f_3(i))^T$. *Then,* (11) *reads*

$$\mathcal{J}_d(u) = \sum_{i=0}^{1} \left(\|f(i) - u(i)\|_2^2 \; + \; \alpha \, \|u(1) - u(0)\|_2 \right)$$

and, in case $d := \|\hat{u}(1) - \hat{u}(0)\|_2 \neq 0$, *the (sub)gradient of* $\mathcal{J}_d$ *at the minimizer* $\hat{u}$ *has to be zero. After some reordering this leads to*

$$\begin{aligned} \hat{u}(0) &= f(0) + \frac{\alpha}{2d}(\hat{u}(1) - \hat{u}(0)), \\ \hat{u}(1) &= f(1) - \frac{\alpha}{2d}(\hat{u}(1) - \hat{u}(0)). \end{aligned}$$

By subtracting these equations and taking the norm of the resulting equation we obtain

$$d = \frac{d}{d+\alpha} \|f(1) - f(0)\|_2$$

and thus $d = \|f(1) - f(0)\|_2 - \alpha$ *if the right-hand side is nonnegative. Consequently, the minimizer of* $\mathcal{J}_d$ *is given by*

$$\begin{aligned} \hat{u}(0) &= f(0) + \frac{\alpha}{2} \frac{f(1) - f(0)}{\|f(1) - f(0)\|_2}, \\ \hat{u}(1) &= f(1) - \frac{\alpha}{2} \frac{f(1) - f(0)}{\|\tilde{f}(1) - f(0)\|_2} \end{aligned} \tag{13}$$

for $\alpha \leq \|f(1) - f(0)\|_2$ *and according to* (12) *by* $\hat{u}(0) = \hat{u}(1) = (f(0) + f(1))/2$ *for larger* α.

We want to compute the minimizer of (11) by SOCP. In this paper, we are only interested in $\mathrm{Sym}_2(\mathbb{R})$. The generalization to $\mathrm{Sym}_n(\mathbb{R})$, $n \geq 3$ is straightforward. We reorder a matrix field $U : \mathbb{Z}_N^2 \to \mathrm{Sym}_2(\mathbb{R})$ into a vector $u \in \mathbb{R}^{3N^2}$ by applying the *vec*-operation

$$u = \begin{pmatrix} \operatorname{vec} U_{11} \\ \operatorname{vec} U_{12} \\ \operatorname{vec} U_{22} \end{pmatrix}, \quad U_{kl} := \big(u_{kl}(i,j)\big)_{i,j=0}^{N-1}, \quad k,l \in \{1,2\}. \tag{14}$$

The (partial) forward difference matrix is defined by $\mathcal{D} = \begin{pmatrix} D_x \\ D_y \end{pmatrix}$ with $D_x = I_N \otimes D$, $D_y = D \otimes I_N$ and

$$D := \begin{pmatrix} -1 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 & 0 \\ & & \ddots & \ddots & \ddots & & \\ 0 & 0 & 0 & \dots & -1 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & -1 & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}. \tag{15}$$

Let 1_N denote the vector consisting of N components 1. Then it is straightforward that minimizing (11) is equivalent to the following SOCP:

$$\begin{gathered} t + \alpha 1_{N^2}^{\mathrm{T}} v \ \to \min \\ \text{s.t.} \begin{pmatrix} \big((1,\sqrt{2},1) \otimes I_{N^2}\big)(f-u) \\ t \\ 1/2 \end{pmatrix} \in \mathcal{K}^{3N^2+2}, \\ \big(u_{11x}, u_{11y}, u_{12x}, u_{12y}, u_{22x}, u_{22y}\big)^{\mathrm{T}} = \big(1_3 \otimes \mathcal{D}\big)u, \\ \left(\begin{pmatrix} u_{11x}(i,j) \\ u_{11y}(i,j) \end{pmatrix}^{\mathrm{T}}, \sqrt{2} \begin{pmatrix} u_{12x}(i,j) \\ u_{12y}(i,j) \end{pmatrix}^{\mathrm{T}}, \begin{pmatrix} u_{22x}(i,j) \\ u_{22y}(i,j) \end{pmatrix}^{\mathrm{T}}, v(i,j) \right)^{\mathrm{T}} \in \mathcal{C}^7, \\ i,j = 0, \dots, N-1, \end{gathered}$$

where $u_{klx}^{\mathrm{T}} = \operatorname{vec}\big(u_{klx}(i,j)\big)_{i,j=0}^{N-1}$.

Remark 4.3. *For the sake of completeness, we mention that there also exists an anisotropic approach* [30] *given by*

$$\mathcal{J}(U) := \int_\Omega \|F - U\|_F^2 + \alpha \operatorname{tr} \Phi\Big(\sum_{j,k=1}^{n} \nabla u_{jk} \nabla u_{jk}^T \Big)\, dxdy. \tag{16}$$

In contrast to (10)*, the function* Φ *is now applied to a matrix now, i.e. to its eigenvalues, and the trace is taken afterwards. This provides the motivation for a novel functional to be introduced in the next section.*

5 New functionals for matrix-fields

Instead of (9) we propose to use the functional

$$\mathcal{J}(U) := \int_\Omega \|F-U\|_F^2 + \alpha \operatorname{tr}\left(\Phi(U_x^2+U_y^2)\right)\, dxdy \tag{17}$$

In contrast to (9), the trace is taken after applying the function Φ to the matrix $U_x^2+U_y^2$. Inspired by (5), we also consider

$$\mathcal{J}(U) := \int_\Omega \|F-U\|_F^2 + \alpha \operatorname{tr}\left(\Phi(U_x^2)+\Phi(U_y^2)\right) dxdy. \tag{18}$$

Again we are only interested in the absolute value function $\Phi(s^2)=|s|$.

The next proposition shows that the functional (17) has an interesting Gâteaux derivative.

Theorem 5.1. *Let Φ be a differentiable function. Then the Euler-Lagrange equations for minimizing the functional* (17) *are given by*

$$\frac{U-F}{\alpha} = \partial_x\left(\Phi'(U_x^2+U_y^2)\bullet U_x\right) + \partial_y\left(\Phi'(U_x^2+U_y^2)\bullet U_y\right). \tag{19}$$

Proof. Let $\varphi(U_x,U_y) := \operatorname{tr}\left(\Phi(U_x^2+U_y^2)\right)$. The Euler-Lagrange equations of (17) are given, for $i,j=1,...,n;\ i\geq j$, by

$$0 = \frac{\partial}{\partial u_{ij}}\|F-U\|_F^2 \;-\; \alpha\left(\frac{\partial}{\partial x}\left(\frac{\partial\varphi}{\partial u_{ijx}}\right)+\frac{\partial}{\partial y}\left(\frac{\partial\varphi}{\partial u_{ijy}}\right)\right).$$

For a scalar-valued function f and an $n\times n$ matrix X, we set $\frac{\partial f(X)}{\partial X} := \left(\frac{\partial f(X)}{\partial x_{ij}}\right)_{i,j=1}^n$. Then, by symmetry of F and U, the Euler-Lagrange equations can be rewritten in matrix-vector form as

$$W_n \circ \frac{U-F}{\alpha} = \frac{1}{2}\left(\frac{\partial}{\partial x}\left(\frac{\partial\varphi}{\partial U_x}\right)+\frac{\partial}{\partial y}\left(\frac{\partial\varphi}{\partial U_y}\right)\right), \tag{20}$$

where W_n denotes the $n\times n$ matrix with diagonal entries 1 and other coefficients 2.

We consider $f(X) := \operatorname{tr}\Phi(X^2)$. Then we obtain by [21, p. 178] and $\operatorname{tr}(A^{\mathrm{T}}B) = (\mathrm{vec}A)^{\mathrm{T}}\mathrm{vec}B$ that

$$\begin{aligned}\mathrm{vec}\frac{\partial f(X)}{\partial X} &= \mathrm{vec}\left(\operatorname{tr}\left(\Phi'(X^2)\,\frac{\partial(X^2)}{\partial x_{ij}}\right)\right)_{i,j=1}^n \\ &= \mathrm{vec}\left((\mathrm{vec}\Psi)^{\mathrm{T}}\mathrm{vec}\frac{\partial(X^2)}{\partial x_{ij}}\right)_{i,j=1}^n\end{aligned}$$

where $\Psi := \Phi'(X^2)$. By [21, p. 182] and since Ψ is symmetric this can be rewritten as

$$\mathrm{vec}\frac{\partial f(X)}{\partial X} = \mathrm{vec}W_n \circ ((I_n \otimes X) + (X \otimes I_n))\,\mathrm{vec}\Psi$$

and using that $\mathrm{vec}(ABC) = (C^{\mathrm{T}} \otimes A)\mathrm{vec}B$ WE INFER THAT

$$\mathrm{vec}\frac{\partial f(X)}{\partial X} = \mathrm{vec}W_n \circ \mathrm{vec}(X\Psi + \Psi X).$$

This implies that

$$\frac{\partial f(X)}{\partial X} = 2\,W_n \circ (\Psi \bullet X). \tag{21}$$

Applying (21) with $f(U_x) := \varphi(U_x, U_y)$ and $f(U_y) := \varphi(U_x, U_y)$, respectively, in (20) we obtain the assertion. □

Univariate matrix–valued functions. We start by considering matrix-valued functions F and U in one spatial variable. In this case, the functionals (17) and (18) coincide and can be written as

$$\mathcal{J}(U) := \int_\Omega \|F - U\|_F^2 + \alpha\,\mathrm{tr}\,|U_x|\,dx \tag{22}$$

with some interval Ω.

Proposition 5.2. i) *The functional* (22) *is strictly convex.*
ii) *For matrices in* $\mathrm{Sym}_2(\mathbb{R})$ *and* $U_x := (u_{jkx})_{j,k=1}^2$, *the functional* (22) *can be rewritten as*

$$\mathcal{J}(U) = \int_\Omega \|F-U\|_F^2 + \alpha\,\max\{\big(4u_{12x}^2 + (u_{11x} - u_{22x})^2\big)^{1/2}, |u_{11x} + u_{22x}|\}\,dx. \tag{23}$$

Proof. i) Let $\lambda : \mathrm{Sym}_n(\mathbb{R}) \to \mathbb{R}^n$ denote the mapping of a matrix to the vector of its eigenvalues in nonincreasing order and let $f(x) := |x_1| + \ldots + |x_n|$. Obviously, f is a symmetric function, i.e., permuting components does not change the function value. Moreover, f is lower semicontinuous and convex. Then, by [6, p. 105], the function $f \circ \lambda$ is also convex. Since the first summand in (22) is strictly convex and the penalizing term coincides with $f \circ \lambda(U_x)$ the whole functional is strictly convex.
ii) Let λ_1 and λ_2 be the eigenvalues of U_x. Then straightforward computation yields

$$\mathrm{tr}\,|U_x| = |\lambda_1| + |\lambda_2| = \big(\mathrm{tr}\,U_x^2 + 2\,|\det U_x|\big)^{1/2}. \tag{24}$$

If $\det U_x = u_{11x}u_{22x} - u_{12x}^2 \geq 0$, then we obtain by (24) that

$$|\lambda_1| + |\lambda_2| = |u_{11x} + u_{22x}| \geq \big((u_{11x} - u_{22x})^2 + 4u_{12x}^2\big)^{1/2}.$$

For $\det U_x < 0$, we get

$$|\lambda_1| + |\lambda_2| = \left((u_{11x} - u_{22x})^2 + 4u_{12x}^2\right)^{1/2} \geq |u_{11x} + u_{22x}|.$$

This implies (23). □

For computations, we consider the discrete counterpart of (23), where we replace the derivative operator by a simple forward difference operator

$$\mathcal{J}_d(U) = \sum_{i=0}^{N-1} \Big(\|F(i) - U(i)\|_F^2 + \alpha \operatorname{tr} |U(i) - U(i-1)| \Big) \tag{25}$$

with $U(-1) := 0$. Unfortunately, the minimizer of (25) does in general not preserve positive definiteness. This is illustrated by the following remark.

Remark 5.3. *We consider the following intuitive example with only two matrices $F(0), F(1) \in \mathrm{Sym}_2^+(\mathbb{R})$. Via the embedding (6) we obtain $F(i) \mapsto f := (f_1(i), f_2(i), f_3(i))^T$. Further, we set $\tilde{f}(i) := (f_1(i), f_2(i))^T$ and similarly for U. Then, (25) reads*

$$\begin{aligned} \mathcal{J}_d(U) \quad = \quad & \sum_{i=0}^{1} \Big(\|\tilde{f}(i) - \tilde{u}(i)\|_2^2 + (f_3(i) - u_3(i))^2 \\ & + \alpha \max\{\|\tilde{u}(1) - \tilde{u}(0)\|_2, |u_3(1) - u_3(0)|\Big). \end{aligned}$$

Let $\left(u_1^*(i), u_2^*(i)\right)^T$, $i = 0, 1$, be the minimizer of

$$\mathcal{J}_{d,1}(\tilde{u}) = \sum_{i=0}^{1} \Big(\|\tilde{f}(i) - \tilde{u}(i)\|_2^2 + \ + \alpha \|\tilde{u}(1) - \tilde{u}(0)\|_2 \Big).$$

Set $u_3^*(i) := f_3(i)$, $i = 0, 1$. Then it is easy to check that in case of

$$\|\tilde{u}^*(1) - \tilde{u}^*(0)\|_2 \geq |u_3^*(1) - u_3^*(0)| \tag{26}$$

the vector field $(u_1^*(i), u_2^*(i), u_3^*(i))^T$, $i = 0, 1$, minimizes $\mathcal{J}_d$. Now the minimizer of $\mathcal{J}_{d,1}(\tilde{u})$ can be computed for $\alpha \leq \|\tilde{f}(1) - \tilde{f}(0)\|_2$ as shown in Remark 4.2. Let $f(0) := (3, 4, 5)^T$, $f(1) := (7, 1, 8)^T \in \mathcal{C}^3$ so that $f(1) - f(0) = (4, -3, 1)^T \notin \mathcal{C}^3$ and $\alpha := 1$. Then, by (13) and (26) the minimizer of $\mathcal{J}_d$ is given by

$$u(0) = f(0) + \frac{1}{10}(4, -3, 0)^T, \quad u(1) = f(1) - \frac{1}{10}(4, -3, 0)^T$$

and $u(0) \notin \mathcal{C}^3$.

By Proposition 5.2 ii), problem (25) can be reformulated as a SOCP. Since this is completely analogeous to (28) in the bivariate case, we formulate the SOCP for the bivariate setting. Positive definiteness of the solution can be ensured by adding the corresponding cone condition in the SOCP. For the example in Remark 5.3 this results in the solution

$$u(0) = f(0) + \frac{1}{10}(3.919, -3.086, 0.131), \quad u(1) = f(1) - \frac{1}{10}(4.008, -2.990, 0).$$

Bivariate matrix–valued functions. The functional (18) can be rewritten as

$$\mathcal{J}(U) = \int_\Omega \|F - U\|_F^2 + \alpha \operatorname{tr}\left(|U_x| + |U_y|\right) dxdy. \tag{27}$$

This functional can be handled similarly as in the univariate case. By Proposition 5.2 and using (14), the corresponding minimization problem can be reformulated as SOCP as follows:

$$\begin{aligned}
&t + \alpha 1_{N^2}^{\mathrm{T}}(v_x + v_y) \ \to \min \\
&\text{s.t.} \begin{pmatrix} \big((1, \sqrt{2}, 1) \otimes I_{N^2}\big)(f - u) \\ t \\ 1/2 \end{pmatrix} \in \mathcal{K}^{3N^2+2}, \\
&\big(u_{11x}, u_{11y}, u_{12x}, u_{12y}, u_{22x}, u_{22y}\big)^{\mathrm{T}} = \big(1_3 \otimes \mathcal{D}\big)u, \\
&(2u_{12x}(i,j), u_{11x}(i,j) - u_{22x}(i,j), v_x(i,j))^{\mathrm{T}} \in \mathcal{C}^3, \\
&(2u_{12y}(i,j), u_{11y}(i,j) - u_{22y}(i,j), v_y(i,j))^{\mathrm{T}} \in \mathcal{C}^3, \\
&(u_{11x}(i,j) + u_{22x}(i,j), v_x(i,j))^{\mathrm{T}} \in \mathcal{C}^2, \\
&(u_{11y}(i,j) + u_{22y}(i,j), v_y(i,j))^{\mathrm{T}} \in \mathcal{C}^2, \quad i,j = 0, \ldots, N-1.
\end{aligned} \tag{28}$$

To ensure positive semi-definiteness of the solution we can simply add the cone condition $(2u_{12}(i,j), u_{11}(i,j) - u_{22}(i,j), u_{11}(i,j) + u_{22}(i,j))^{\mathrm{T}} \in \mathcal{C}^3$ to (28).

The functional (17) can be rewritten as

$$\mathcal{J}(U) = \int_\Omega \|F - U\|_F^2 + \alpha \operatorname{tr} \sqrt{U_x^2 + U_y^2}\, dxdy \tag{29}$$

and in particular in case $\mathrm{Sym}_2(\mathbb{R})$ as

$$\mathcal{J}(U) = \int_\Omega \|F - U\|_F^2 + \alpha \sqrt{\eta}\, dxdy,$$

where $\eta = \eta(u_{11x}, u_{12x}, u_{22x}, u_{11y}, u_{12y}, u_{22y})$ is given by

$$\eta = u_{11x}^2 + 2u_{12x}^2 + u_{22x}^2 + u_{11y}^2 + 2u_{12y}^2 + u_{22y}^2$$
$$+\ 2\left((u_{11x}u_{22x} - u_{12x}^2)^2 + (u_{11y}u_{22y} - u_{12y}^2)^2 + (u_{11x}u_{22y} - u_{12x}u_{12y})^2\right.$$
$$\left.+(u_{11y}u_{22x} - u_{12x}u_{12y})^2 + (u_{11x}u_{12y} - u_{12x}u_{11y})^2 + (u_{12y}u_{22x} - u_{12x}u_{22y})^2\right)^{1/2}.$$

To compute a minimizer of (29) we apply Theorem 5.2 and solve the corresponding reaction–diffusion equation for $t \to \infty$

$$U_t = F - U + \alpha\left(\partial_x\left(\Phi'(U_x^2 + U_y^2) \bullet U_x\right) + \partial_y\left(\Phi'(U_x^2 + U_y^2) \bullet U_y\right)\right)$$

with Φ as in (4), homogeneous Neumann boundary conditions and initial value F by a difference method. More precisely, we use the iterative scheme

$$U^{(k+1)} = (1-\tau)U^{(k)} + \tau F + \tau\alpha\left(\partial_x\left(G^{(k)} \bullet U_x^{(k)}\right) + \partial_y\left(G^{(k)} \bullet U_y^{(k)}\right)\right)$$

with sufficiently small time step size τ and $G^{(k)} := \Phi'((U_x^{(k)})^2 + (U_y^{(k)})^2)$. The inner derivatives including those in G were approximated by forward differences and the outer derivatives by backward differences so that the penalizing term becomes

$$\frac{1}{h_1}\left(G(i,j) \bullet \frac{U(i+1,j) - U(i,j)}{h_1} - G(i-1,j) \bullet \frac{U(i,j) - U(i-1,j)}{h_1}\right)$$
$$+\quad \frac{1}{h_2}\left(G(i,j) \bullet \frac{U(i,j+1) - U(i,j)}{h_2} - G(i,j-1) \bullet \frac{U(i,j) - U(i,j-1)}{h_2}\right),$$

where h_i, $i = 1, 2$ denote the pixel distances in x and y–direction. Alternatively, we have also worked with symmetric differences for the derivatives. Then we have to replace e.g. $G(i,j)$ in the first summand by $\tilde{G}(i+1,j) + \tilde{G}(i,j))/2$ and $\tilde{G}$ is now computed with symmetric differences.

6 Numerical Results

Finally, we present some numerical results demonstrating the performance of the various methods. All algorithms were implemented in MATLAB. Moreover, we have used the software package MOSEK for SOCP. We restrict our attention to $\mathrm{Sym}_2(\mathbb{R})$.

We start with the 1D matrix–valued function in Fig. 2. To all components of the original data in $[0,1]$ we added white Gaussian noise with standard deviation 0.1. We computed the minimizer of the Deriche-Tschumperlé functional (9) (left) and of our new functional (23) (right) by SOCP. The bottom of the figure shows the ℓ_2–norm (of three matrix components) and the Frobenius norm of the difference betweenof the original and the denoised signal $(\sum \|F(i) - \hat{U}(i)\|_F^2)^{1/2}$ in dependence on the regularization parameter α. Note that the shape of the curve and its minimal point does not change

if we use $\sum \|F(i) - \hat{U}(i)\|_F$ instead. The actual minima w.r.t. the Frobenius norm are given by $\alpha = 0.8$ and min $= 0.2665$ for (9) and $\alpha = 0.8$ and min $= 0.2276$ for (23). The denoised signals corresponding to the smallest error in the Frobenius–norm are depicted in the middle of the figure. It appears that the new method performs slightly better w.r.t. these error norms. The visual results confirm this impression. The larger ellipses obtained by the first method (9) slightly overlap while there are gaps between the smaller ones. We do not have this effect for the minimizer of (23) at the left–hand side.

Now we turn to 2D matrix–valued functions. We compare the minimizer of the Deriche-Tschumperlé functional (9) with those of our new functionals (27) and (29) For the first two functionals we applied SOCP while the third one was computed via the reaction–diffusion equation with time step size $\tau = 0.00025$. The iterations were stopped when the relative error in the ℓ_2-norm between two consecutive iterations became smaller than 10^{-8} (approximately 20000 iterations) although the result remains visually static much earlier.

In Fig. 3 we added white Gaussian noise with standard deviation 0.1 to all components of the original data. The bottom of the figure contains again the error plots. The actual minima w.r.t. the Frobenius norm are given by $\alpha = 0.28$ and min $= 0.7128$ for (9), $\alpha = 0.18$ and min $= 0.6489$ for (27) and $\alpha = 0.18$ and min $= 0.7426$ for (29). Regarding these errors, method (27) performs best, however visually it is hard to distinguish between the methods.

Our third example in Fig. 4 is similar to the second one except that we have to apply another visualization technique based on OpenGL for the larger matrix-field. To all components of the original data in [0,2] we added white Gaussian noise with standard deviation 0.6. We use the same parameters as in Fig. 3. The bottom of the figure contains the error plots for the three methods. The actual minima w.r.t. the Frobenius norm are given by $\alpha = 1.75$ and min $= 12.19$ for (9), $\alpha = 1.15$ and min $= 11.6$ for (27) and $\alpha = 1.2$ and min $= 10.79$ for (29). With respect to the computed errors the new methods outperform the one based on the Deriche-Tschumperlé functional, where the third method performs best.

Finally, we remark that we have restricted our attention to small artificial examples to see some differences between the various methods. In general it is no problem to use SOCP for matrix–valued images of size e.g. 128×128.

References

[1] A. Barvinok. *A Course in Convexity, Graduate Studies in Mathematics.* AMS, Providence, RI, 2002.

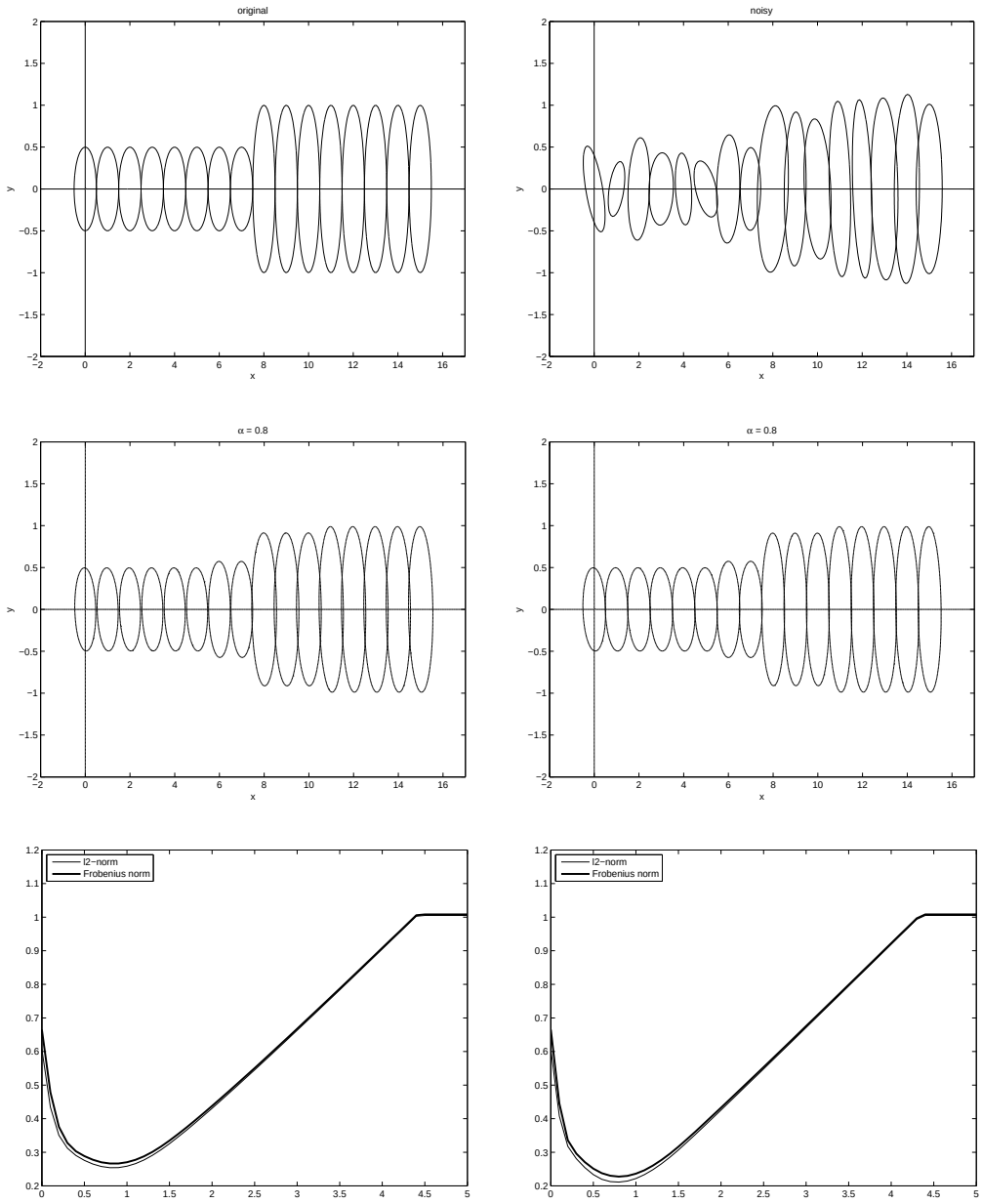

Figure 2: Denoising of a matrix–valued signal. **Top**: Original signal (left), noisy signal (right). **Middle**: Denoised image for α corresponding to the smallest error in the Frobenius norm for (9) and (23) (left to right). **Bottom**: l_2–error and error of the Frobenius norm in dependence on the regularization parameter α for the minimizers of (9) and (23) (left to right).

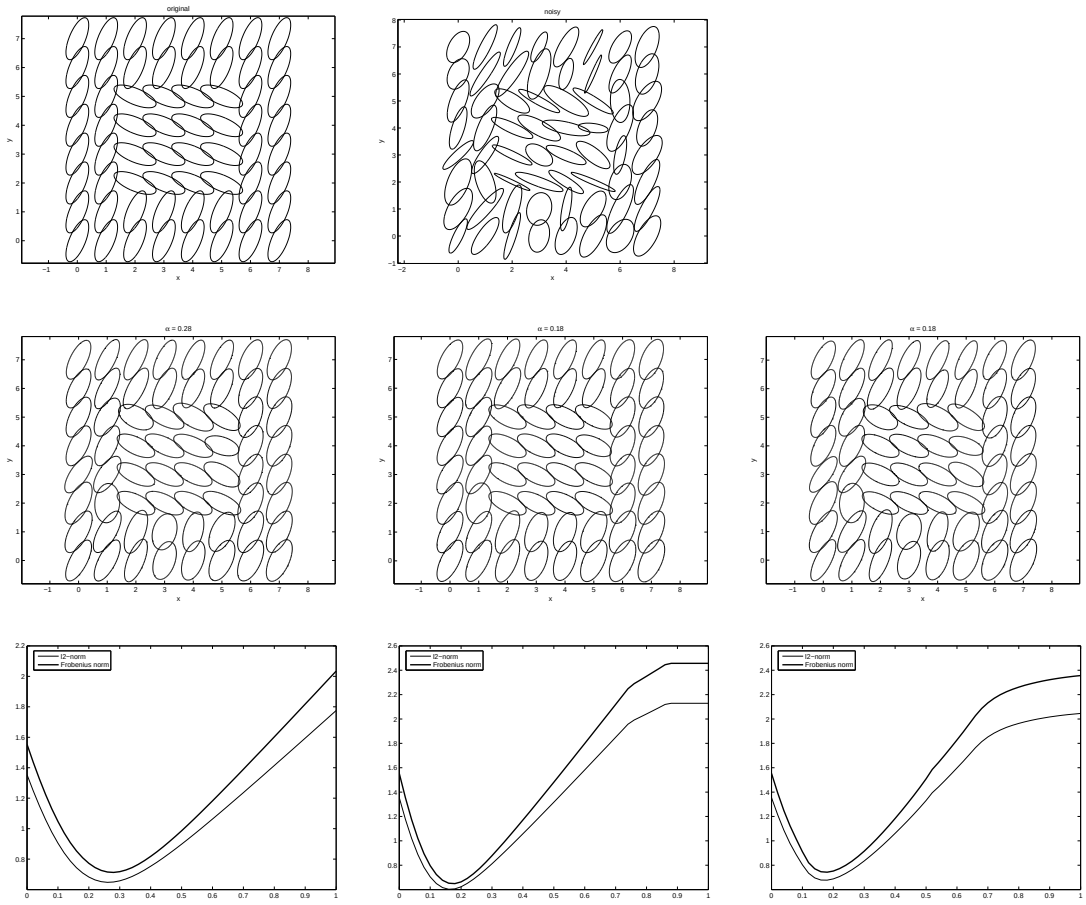

Figure 3: Denoising of a matrix–valued image. **Top**: Original signal (left), noisy signal (right). **Middle**: Denoised image for α corresponding to the smallest error in the Frobenius norm for (9), (27) and (29) (left to right). **Bottom**: l_2–error and error of the Frobenius norm in dependence on the regularization parameter α for the minimizers of (9), (27) and (29) (left to right).

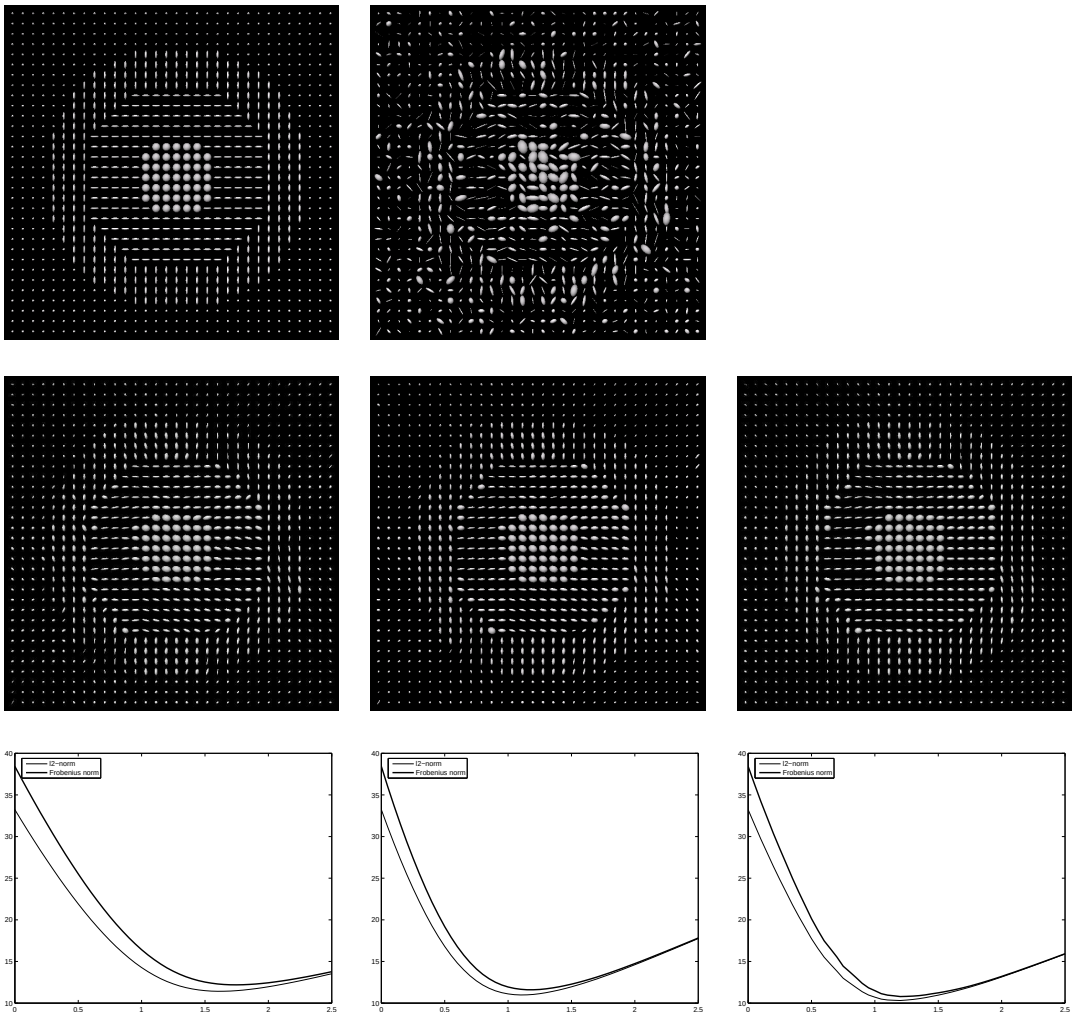

Figure 4: Denoising of a matrix–valued image. **Top**: Original signal (left), noisy signal (right). **Middle**: Denoised image for α corresponding to the smallest error in the Frobenius norm for (9), (27) and (29) (left to right). **Bottom**: l_2–error and error of the Frobenius norm in dependence on the regularization parameter α for the minimizer of (9), (27) and (29) (left to right).

[2] P. J. Basser. Inferring microstructural features and the physical state of tissues from diffusion-weighted images. *Nuclear Magnetic Resonance in Biomedicine*, 8:333–334, 1995.

[3] P. J. Basser, J. Mattiello, and D. LeBihan. MR diffusion tensor spectroscopy and imaging. *Biophysical Journal*, 66:259–267, 1994.

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

[5] J. Bigün, G. H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, Aug. 1991.

[6] J. M. Borwein and A. S. Lewis. *Convex Analysis and Nonlinear Optimization*. Springer, New York, 2000.

[7] T. Brox, J. Weickert, B. Burgeth, and P. Mrázek. Nonlinear structure tensors. *Image and Vision Computing*, 24(1):41–55, 2006.

[8] B. Burgeth, A. Bruhn, S. Didas, J. Weickert, and M. Welk. Morphology for matrix-data: Ordering versus PDE-based approach. *Image and Vision Computing*, 2007, in print.

[9] B. Burgeth, A. Bruhn, N. Papenberg, M. Welk, and J. Weickert. Mathematical morphology for matrix fields induced by the Loewner ordering in higher dimensions. *Signal Processing*, 87(2):277–290, 2007.

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

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

[12] P. G. Ciarlet. *Introduction to Numerical Linear Algebra and Optimisation*. Cambridge University Press, Cambridge, 1989.

[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] D. Goldfarb and W. Yin. Second order cone programming methods for total variation–based image restoration. *SIAM J. Scientific Computing*, 2(27):622–645, 2005.

[15] G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer, Dordrecht, 1995.

[16] C. G. Harris and M. Stephens. A combined corner and edge detector. In *Proc. Fouth Alvey Vision Conference*, pages 147–152, Manchester, England, Aug. 1988.

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

[18] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, UK, 1990.

[19] M. Lysaker, A. Lundervold, and X. Tai. Noise removal using fourth-order partial differential equations with applications to medical magnetic resonance images in space and time. Technical Report CAM-02-44, Department of Mathematics, University of California at Los Angeles, CA, U.S.A., 2002.

[20] S. B. M. S. Lobo, L. Vandenberghe and H. Lebret. Applications of second order cone programming. *Linear Algebra and its Applications*, 1998.

[21] J. R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. J. Wiley and Sons, Chichester, 1988.

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

[23] A. R. Rao and B. G. Schunck. Computing oriented texture fields. *CVGIP: Graphical Models and Image Processing*, 53:157–185, 1991.

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

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

[26] G. Steidl, M. Welk, and J. Weickert. Locally analytic schemes: a link between diffusion filtering and wavelet shrinkage. *Applied and Computational Harmonic Analysis*, page to appear, 2007.

[27] D. Tschumperlé and R. Deriche. Diffusion tensor regularization with contraints preservation. In *Proc. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 948–953, Kauai, HI, 2001. IEEE Computer Science Press.

[28] C. R. Vogel. *Computational Methods for Inverse Problems.* SIAM, Philadelphia, 2002.

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

[30] J. Weickert and T. Brox. Diffusion and regularization of vector- and matrix-valued images. In M. Z. Nashed and O. Scherzer, editors, *Inverse Problems, Image Analysis, and Medical Imaging*, volume 313 of *Contemporary Mathematics*, pages 251–268. AMS, Providence, 2002.