SIAM J. Imaging Sciences
Vol. 7, No. 4, pp. 2226–2257

Total Variation Regularization for Manifold-Valued Data[*]

Andreas Weinmann[†], Laurent Demaret[†], and Martin Storath[‡]

Abstract. We consider total variation (TV) minimization for manifold-valued data. We propose a cyclic proximal point algorithm and a parallel proximal point algorithm to minimize TV functionals with ℓ^p-type data terms in the manifold case. These algorithms are based on iterative geodesic averaging which makes them easily applicable to a large class of data manifolds. As an application, we consider denoising images which take their values in a manifold. We apply our algorithms to diffusion tensor images and interferometric SAR images as well as sphere- and cylinder-valued images. For the class of Cartan–Hadamard manifolds (which includes the data space in diffusion tensor imaging) we show the convergence of the proposed TV minimizing algorithms to a global minimizer.

Key words. total variation minimization, manifold-valued data, proximal point algorithm, diffusion tensor imaging

AMS subject classifications. 65K05, 65K10, 68U10, 94A08

DOI. 10.1137/130951075

1. Introduction.

Data taking values in a manifold appear naturally in various signal and image processing applications. One example is diffusion tensor imaging, where the data live in the Riemannian manifold of positive (definite) matrices; see, e.g., [11, 69]. Other examples are color images based on nonflat color models [23, 84, 57, 58]. Here the data has circle- or sphere-valued components. Data with values on the circle also appear in the context of interferometric SAR images [60]. SO(3) and motion group-valued data were considered in [83, 73].

Processing manifold-valued data has gained a lot of interest in recent years. To mention only some examples, wavelet-type multiscale transforms for manifold data have been considered in [83, 50, 86]. Manifold-valued partial differential equations are the subject of study in the papers [81, 25, 49]. Furthermore, statistical issues on Riemannian manifolds are the topic of [38, 39, 40, 67, 14, 15, 68].

The present paper deals with total variation (TV) minimization for data taking values in a manifold. Our main application is the denoising of such manifold-valued data. For scalar data, TV minimization was shown to be a powerful tool for edge-preserving denoising [75].

[*] Received by the editors December 30, 2013; accepted for publication (in revised form) July 29, 2014; published electronically November 18, 2014. This work was supported by the German Federal Ministry for Education and Research under SysTec grant 0315508.

http://www.siam.org/journals/siims/7-4/95107.html

[†] Helmholtz Center Munich, 85764 Oberschleißheim, Germany, and Department of Mathematics, Technische Universität München, Germany (andreas.weinmann@tum.de, laurent.demaret@helmholtz-muenchen.de). The first author's research was supported by the Helmholtz Association within the young investigator group VH-NG-526.

[‡] Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland (martin.storath@epfl.ch). This author's research was supported by the European Research Council (ERC) under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement 267439.

For images, the anisotropic version of TV minimization is given by

$$\operatorname*{argmin}_{x\in M^{n\times m}} \left\{ \frac{1}{p}\sum_{i,j=1}^{n,m} d(x_{ij},f_{ij})^p + \alpha \sum_{i,j=1}^{n-1,m} d(x_{ij},x_{i+1,j}) + \alpha \sum_{i,j=1}^{n,m-1} d(x_{ij},x_{i,j+1}) \right\}. \tag{1.1}$$

Here, f_{ij} is the observed data at grid location (i,j) and x is the argument to optimize for yielding the regularized value x_{ij} at the corresponding grid location. For scalar data, the symbol $d(y,z)=|y-z|$ simply denotes the Euclidean distance between y and z. The distance to data f is measured in the ℓ^p norm. For Gaussian noise, $p=2$ is reasonable, whereas, for noise types with heavier tails such as Laplacian noise, $p=1$ is more natural. The natural generalization of the TV problem for data on a manifold M is given by using the distance d induced by the Riemannian metric on the manifold instead of the Euclidean distance in (1.1).

We introduce algorithms for the ℓ^p-TV problem for general Riemannian manifold data. Furthermore, we show convergence towards a (global) minimizer for the class of Cartan–Hadamard manifolds of which the data space in diffusion tensor imaging is a particular member. Our experiments show the denoising capability of ℓ^p-TV minimization in the manifold context.

1.1. TV regularization for scalar-valued, vector-valued, and matrix-valued data.

TV regularization was first introduced in the early 1990s by Rudin, Osher, and Fatemi [75]. A central advantage of TV regularization compared with classical Tikhonov regularization is that it preserves sharp edges [79, 47]. Especially due to this property, TV minimization has been used in a vast amount of applications. Examples are biomedical imaging [31], geophysics [3], and computer vision [89, 27], to mention only a few.

Theoretical properties of TV regularization have been investigated in a series of papers. For instance, results on existence and uniqueness of minimizers have been proved in [20]. Connections to wavelet shrinkage are shown in [70], and equivalences between diffusion techniques, thresholding strategies, and TV minimization can be found in [77].

A lot of different algorithms for TV minimization of scalar- and vector-valued images have been proposed in the last 20 years. In their original work, Rudin, Osher, and Fatemi [75] consider ℓ^2 data terms. They use gradient descent on the Euler–Lagrange equations of the (scalar-valued) TV functional. Further methods are based on Fenchel duals [19], the alternating direction method of multipliers [88], and split Bregman methods [46].

Several authors have studied the TV problem with ℓ^1 data terms [2, 63, 22]. Approaches based on the ℓ^1-TV functional enjoy the edge-preserving properties of TV regularizers while, in addition, being more robust to outliers. Various solution strategies have been proposed for the ℓ^1-TV problem. To mention some examples, schemes based on smooth approximations are presented in [64, 65]; semismooth Newton method–based approaches are the topic of [28]; and primal-dual methods are proposed in [21, 33].

TV regularization for matrix-valued images has recently been considered in [74, 58]. There, the data manifolds are embedded to a higher-dimensional Euclidean space and the metric in (1.1) is the Euclidean metric of the ambient space. The minimization problem is then split into a TV minimization on Euclidean space and a projection step to the data manifold which are applied alternatingly.

1.2. Proposed algorithms for TV minimization for manifold-valued data. In this paper we derive algorithms for the TV problem (1.1) on (Riemannian) manifold-valued data. Our algorithms are based on iterative geodesic averaging. More precisely, we decompose the TV functional in (1.1) into a sum of functionals in such a way that we are able to explicitly compute the proximal mappings of these functionals on the general (Riemannian) manifold. For general (Riemannian) manifolds, we obtain that these proximal mappings are given in terms of points on certain geodesics. So, in order to make the algorithms work on a concrete manifold, the only operations we need are those needed for calculating geodesics. The spaces which frequently occur as data spaces are matrix groups or related symmetric spaces. So usually, there are explicit formulas available for this task.

Our algorithms are iterative schemes. In each iteration, we apply the above proximal mappings of the functionals decomposing the TV functional. The first algorithm is a *cyclic proximal point algorithm*. This means that we successively apply the proximal mappings using the output related to the ith summand as a new input for the proximal mapping of the $(i+1)$st summand. The second algorithm is a *parallel proximal point algorithm*. Here the proximal mappings are calculated for the same initial point and averaged afterwards. Since computing mean values on a manifold is a relatively expensive iterative procedure, we also consider a variant which does only approximative averaging (but yields comparable results). We call this variant a *fast parallel proximal point algorithm*. Due to the averaging procedure, the parallel algorithms need more operations in total. However, they have higher potential for parallelization.

Our algorithms belong to the class of proximal splitting methods (for manifold-valued data). A survey on proximal splitting methods (for scalar data) is [29]. Section 7 of the paper [29] also describes a related parallel algorithm. Parallel proximal point algorithms were also considered in [16]. Cyclical proximal point algorithms have been studied in [13] (for linear spaces) and in [9], where they are applied for the computation of means and medians in Hadamard spaces.

Principally, our algorithms work for all ℓ^p data terms with $p \geq 1$ as well as for regularizing terms based on qth variation, $q \geq 1$, instead of total variation. This in particular includes the classical Tikhonov regularization which corresponds to $p = q = 2$. For $p, q = 1, 2$ we give nice closed form expressions; in the other cases, one needs the numerical solution of a certain nonlinear equation. We furthermore consider Huber data terms as well as Huber regularizing terms (which are sometimes called "Huber-ROF"). The latter are employed to avoid undesired staircasing effects; see [21].

For the class of Cartan–Hadamard manifolds, we obtain the convergence of our geodesic averaging–based schemes towards a (global) minimizer of (1.1). Cartan–Hadamard manifolds are Riemannian manifolds containing many symmetric spaces, such as the data space in diffusion tensor imaging. Our convergence statements also hold true for the more general class of Hadamard spaces and for regularization based on qth variation as well as Huber data and regularization terms.

1.3. Applications. We demonstrate the denoising capabilities of our algorithms on various data spaces. It is the common observation of all experiments that the TV approach reliably removes noise from manifold-valued data while preserving edges.

First, we consider diffusion tensor images. Diffusion tensor imaging (DTI) is a technique to quantify noninvasively the diffusional characteristics of a specimen [11, 52]. Here the underlying data space is the space of positive matrices. According to the model, the diffusivity in direction v is determined by $v^T Av$, where A is a positive matrix representing data. The space of positive matrices becomes a Cartan–Hadamard manifold when equipped with a suitable Riemannian metric. This means that our algorithm provably converges to a minimizer in the DTI setup. We demonstrate the denoising performance with a real diffusion tensor image of a human brain and with synthetic data; see Figures 1 and 2.

Next, we consider interferometric synthetic aperture radar (InSAR) data. InSAR is an important airborne imaging modality for geodesy [60]. In our concrete example, the InSAR image has the interpretation of a wrapped periodic version of a digital elevation model [72]. Hence the underlying data space is the sphere S^1. From the experiment in Figure 4 we see that TV minimization is capable of removing almost all the noise from the InSAR image. We further observe that ℓ^1 and Huber data terms are slightly more robust to outliers in the data than the ℓ^2 data term.

Our third application is image denoising in nonlinear color spaces. We consider the LCh space which consists of real-valued luminance and chromaticity components L and C as well as an S^1-valued hue component h. Thus the underlying manifold is the cylinder $\mathbb{R}^2 \times S^1$. We note that, although the underlying manifold $\mathbb{R}^2 \times S^1$ is a product space, the algorithms cannot be applied separately to the components. In our experiment we obtain a better reconstruction quality by manifold-valued TV minimization in the LCh color space than by classical vectorial TV minimization in the standard RGB space; see Figure 5.

We continue with the sphere S^2. Data with values in S^2 appear in, e.g., chromaticity-based image processing [23] and as orientation fields of three-dimensional (3D) images [71]. We here consider a synthetic example on which we impose von Mises–Fisher noise. Figure 6 shows that, also for data with values in the sphere S^2, the noise is almost perfectly removed and that the edges are not smoothed out.

We conclude with the rotation group SO(3) as data spaces. Data with values in SO(3) appear, for example, in the context of aircraft orientations [83], protein alignments [48], and the tracking of 3D rotational data arising in robotics [34]. We apply our methods to a synthetic time series on which we imposed noise based on a matrix Fisher distribution. The results confirm their denoising capability, and they also reveal that a Huber regularizing term is less affected by staircasing effects; cf. Figure 7.

1.4. Organization of the paper.

We start out by developing algorithms for TV minimization for manifold-valued data in section 2. Then we show the convergence of our algorithms towards (global) minimizers of the TV minimization problem (1.1) in Hadamard spaces in section 3. In section 4, we apply our algorithms to denoising data with values in concrete manifolds.

2. Algorithms for TV minimization for manifold-valued data.

In the following we propose two algorithms for TV minimization for data which take their values in a manifold. We consider ℓ^1 and ℓ^2 as well as Huber data terms. Our algorithms are based on iterative geodesic averaging. The appearing geodesic averages are the minimizers of certain proximal mappings which arise as follows: we split the TV functional into basic building blocks and consider the proximal mappings of these building blocks. The first algorithm performs the iteration of the proximal mappings in a cyclical way, whereas the second does so in a parallel way.

2.1. Splitting of the TV functional and proximal mappings. Let us consider the problem of (bivariate) ℓ^p-TV^q minimization

$$\frac{1}{p}\sum_{i,j} d^p(x_{ij}, f_{ij}) + \alpha\frac{1}{q}\sum_{i,j} d^q(x_{ij}, x_{i+1,j}) + \alpha\frac{1}{q}\sum_{i,j} d^q(x_{ij}, x_{i,j+1}) \to \text{min}. \tag{2.1}$$

The data f_{ij} as well as the arguments x_{ij} to minimize take their values in a Riemannian manifold M. The symbol d^p denotes the pth power of the distance induced by the Riemannian metric. For a bounded (complete) Riemannian manifold, the functional obviously has a minimizer since continuous functions have minima on compact sets. For the unbounded case, we notice that going too far away from the data f leads to high functional values. Hence the set of minimizer candidates is actually confined to a bounded set which brings us back to the already discussed situation and minimizers exist.

Setting $q = 1$ in (2.1), we get the discrete (anisotropic) TV functional with ℓ^p data term. In particular, if $p = 1$, we are in the ℓ^1-TV setting. The case $q = 2$ corresponds to the classical Tikhonov regularization term in the scalar case. We comment on Huber data and regularizing terms in section 2.5.

We here consider a discrete domain variant of the TV functional. An analysis of continuous domain TV-type functionals for manifold-valued data has been carried out in [45, 44, 43].

Our approaches towards the minimization of (2.1) are based on rewriting (2.1) as a sum of simpler functions. We consider the function $F : M \times \cdots \times M \to \mathbb{R}$, which is the data term given by

$$F(x) = \frac{1}{p}\sum_{i,j=1}^{n,m} d^p(x_{ij}, f_{ij}), \tag{2.2}$$

as well as, for i, j, the functions $G_{ij}, H_{ij} : M \times \cdots \times M \to \mathbb{R}$ given by

$$\begin{aligned} G_{ij}(x) &= \frac{1}{q} d^q(x_{ij}, x_{i,j+1}), \\ H_{ij}(x) &= \frac{1}{q} d^q(x_{ij}, x_{i+1,j}). \end{aligned} \tag{2.3}$$

Using this notation, the minimization problem (2.1) has the form

$$F(x) + \alpha\sum_{i,j} G_{ij}(x) + \alpha\sum_{i,j} H_{ij}(x) \to \text{min}. \tag{2.4}$$

For each summand in (2.4), we consider its proximal mapping [62, 35, 8]. The proximal mappings of the G_{ij} are defined by the minimization problem

$$\text{prox}_{\lambda G_{ij}} x = \underset{y \in M^{n\times m}}{\text{argmin}} \left(\lambda G_{ij}(y) + \frac{1}{2} d^2(x, y)\right), \tag{2.5}$$

where the parameter $\lambda > 0$ and the distance d on the product manifold $M^{n\times m}$ is given by $d^2(x,y) = \sum_{i,j=1}^{n,m} d(x_{ij},y_{ij})^2$. The proximal mappings of F and the H_{ij} are defined analogously. The crucial point is that, using the splitting (2.4), the proximal mappings of all appearing summands can be explicitly computed as geodesic averages. More precisely, solving the minimization problem of (2.5) reduces to computing points on shortest geodesics joining given points. The same is true for the analogous problems for F and the H_{ij}.

The generalization of our approach to more than two-dimensional (2D) (discrete) domains is straightforward. It is done by introducing new functionals analogous to $G_{i,j}, H_{ij}$—one for each dimension. In three dimensions, for example, one would consider three analogously defined functionals $G_{ijk}, H_{ijk}, I_{ijk}$ with three indices i,j,k—one for each dimension of the domain. Starting from these functionals, the derivation of corresponding algorithms follows the same scheme as worked out for the 2D case in the following.

A heuristic derivation of the proximal mappings. For illustrative and readability purposes, we first give a heuristic derivation of the proximal mappings of the functionals G_{ij}, H_{ij}, and F in the following paragraph. In the subsequent paragraph a precise mathematical statement concerning these proximal mappings, namely Proposition 1, is formulated. Being informal in this heuristic derivation, we tacitly assume in this paragraph that the points in the Riemannian manifold are sufficiently near each other so that the following arguments apply. Let us consider the proximal mapping of G_{ij} given by (2.5). More precisely, we consider the mapping $y \to \lambda G_{ij}(y) + \frac{1}{2}d^2(x,y)$ in (2.5) which is defined on $M \times \cdots \times M$. This means that we consider all members of y, in particular y_{ij} and its neighbors, as varying. A necessary condition for $x' \in M^{n\times m}$ to be a minimizer of this mapping is that 0 is in the (sub)gradient of $\lambda G_{ij}(x') + \frac{1}{2}d^2(x,x')$. This immediately implies that, for the (k,l)th component of the proximal mapping $\mathrm{prox}_{\lambda G_{ij}} x$,

$$(\mathrm{prox}_{\lambda G_{ij}} x)_{kl} = x_{kl} \quad \text{for} \quad k \neq i, \text{ and } l \neq j, j+1. \tag{2.6}$$

For the indices $k = i$ and $l = j, j+1$ we use that the gradient of the mapping $M \to \mathbb{R}$, $z \mapsto d^p(z,v)/p$ fulfills (see, e.g., [1, eq. (2.8)])

$$\nabla_z d^p(z,v)/p = \frac{\exp_z^{-1}(v)}{d^{2-p}(z,v)}. \tag{2.7}$$

We consider the member y_{ij} and its neighbor $y_{i,j+1}$ (both as varying) in the mapping $(y_{ij}, y_{i,j+1}) \to G_{ij}(y) + \frac{1}{2}d^2(x,y)$. Applying (2.7) we get as necessary conditions for a minimizer x' in (2.5) that

$$\begin{aligned} \lambda \tfrac{1}{d^{2-p}(x'_{ij},x'_{i,j+1})} \exp^{-1}_{x'_{ij}}(x'_{i,j+1}) + \exp^{-1}_{x'_{ij}}(x_{ij}) = 0,\\ \lambda \tfrac{1}{d^{2-p}(x'_{ij},x'_{i,j+1})} \exp^{-1}_{x'_{i,j+1}}(x'_{ij}) + \exp^{-1}_{x'_{i,j+1}}(x_{i,j+1}) = 0. \end{aligned}$$

Here $\exp_z^{-1}$ denotes the inverse of the Riemannian exponential mapping at the point z. Thus, both summands in the first condition are tangent vectors at x'_{ij}. They point in opposite directions, and so the first condition implies that the three points $x'_{i,j}$, $x'_{i,j+1}$, and $x_{i,j}$ lie on a common geodesic in M. Analogously, the second condition implies that the three points

$x'_{i,j}$, $x'_{i,j+1}$, and $x_{i,j+1}$ also lie on a common geodesic. Hence, the four points must lie on one geodesic. In particular, the points $x'_{i,j}$, $x'_{i,j+1}$—which are the (i,j)th and $(i,j+1)$st components of the proximal mapping of G_{ij} applied to x—lie on the geodesic joining $x_{i,j}$ and $x_{i,j+1}$.

Then, after some technical considerations (cf. the proof of Theorem 2), the locations of the points $x'_{i,j} = (\mathrm{prox}_{\lambda G_{ij}} x)_{i,j}$ and $x'_{i,j+1} = (\mathrm{prox}_{\lambda G_{ij}} x)_{i,j+1}$ are explicitly given as follows. We have

$$\begin{aligned} (\mathrm{prox}_{\lambda G_{ij}} x)_{ij} &= [x_{ij}, x_{i,j+1}]_t, \\ (\mathrm{prox}_{\lambda G_{ij}} x)_{i,j+1} &= [x_{i,j+1}, x_{ij}]_t, \end{aligned} \tag{2.8}$$

where the symbol $[\cdot,\cdot]_t$ denotes the point reached after time t on the unit speed geodesic starting at the first argument in the direction of the second argument. In the TV case ($q = 1$), we have

$$t = \begin{cases} \lambda & \text{if } \lambda < \frac{1}{2} d(x_{ij}, x_{i,j+1}), \\ d(x_{ij}, x_{i,j+1})/2 & \text{else.} \end{cases} \tag{2.9}$$

For $q = 2$, which corresponds to quadratic variation, we get

$$t = \frac{\lambda}{1+2\lambda} d(x_{ij}, x_{i,j+1}). \tag{2.10}$$

The proximal mappings of the H_{ij} are obtained in a completely analogous way. It remains to find the proximal mapping of F, which means finding the proximal mapping of the distance function in M. This is well known and can be found, e.g., in [35]. They can again be written as geodesic averages and are explicitly given by

$$(\mathrm{prox}_{\lambda F})_{ij}(x) = [x_{ij}, f_{ij}]_t, \tag{2.11}$$

where, for the ℓ^2 data term,

$$t = \frac{\lambda}{1+\lambda} d(x_{ij}, f_{ij}). \tag{2.12}$$

For the ℓ^1 data term,

$$t = \begin{cases} \lambda & \text{if } \lambda < d(x_{ij}, f_{ij}), \\ d(x_{ij}, f_{ij}) & \text{else.} \end{cases} \tag{2.13}$$

This corresponds to the equivalent of *soft thresholding* in the context of manifolds.

A mathematically rigorous statement on the proximal mappings for general Riemannian manifolds. After having heuristically obtained formulas for the proximal mappings of the functionals G_{ij}, H_{ij}, and F, let us now formulate a mathematically precise statement under which conditions these formulas describe the corresponding proximal mappings. To this end, we need some preparation. We consider the functional G_{ij} with fixed i,j (analogously for $H_{i,j}$ and F). For given x, we let Y^* be given by

$$Y^* = \operatorname*{argmin}_{y \in M^{n\times m}} \left(\lambda G_{ij}(y) + \frac{1}{2} d^2(x,y) \right). \tag{2.14}$$

This means that Y^* is the set of (global) minimizers of the minimization problem related to the proximal mapping (2.5). We furthermore define the set Y' by

$$\begin{aligned} Y' = \{y \in M^{n\times m} : \; & y_{k,l} = x_{k,l} \text{ for } k \neq i \text{ and } l \neq j, j+1, \\ & y_{i,j} = [x_{i,j}, x_{i,j+1}]^\gamma_t, y_{i,j+1} = [x_{i,j+1}, x_{i,j}]^\gamma_t \text{ for some} \\ & \text{shortest geodesic } \gamma \text{ in unit speed parametrization}\} \end{aligned} \tag{2.15}$$

with t given by (2.9) and (2.10), respectively. This means that Y' contains an element y for every shortest geodesic joining $x_{i,j}$ and $x_{i+1,j}$. The symbol $[x_{i,j}, x_{i,j+1}]^\gamma_t$ denotes the point on the unit speed geodesic γ (which is a shortest path) starting at $x_{i,j}$ reached after time t. Now we relate Y^* and Y'.

Proposition 1. *Let M be a complete connected Riemannian manifold. Then the minimizers Y^* of the proximal mapping related minimization problem for the functional G_{ij} and input x in* (2.5) *are precisely the members of Y' given by* (2.15). *Analogous statements hold for the H_{ij} and F.*

In particular, if there is only one shortest geodesic joining the data items $x_{i,j}, x_{i,j+1}$ (which is always the case for nearby points and almost everywhere globally), then the proximal mapping of G_{ij} is well defined (as a single-valued mapping). It is given by (2.6), (2.8). *Analogous statements hold for the H_{ij} and F.*

The proof of Proposition 1 is given in Appendix A.

2.2. A cyclic proximal point algorithm for TV minimization for manifold-valued data.

The first algorithm we propose for TV minimization for manifold-valued data is a cyclic proximal point algorithm based on geodesic averaging. For vector space data, cyclical proximal point algorithms were considered in [13]. For Hadamard spaces, they were investigated by Bačák [9], who applied them to the computation of means and medians.

We now derive a cyclic proximal point algorithm for the minimization of the ℓ^p-TV^q functional (2.1). We consider the problem in the form $F(x) + \alpha \sum_{i,j} G_{ij}(x) + \alpha \sum_{i,j} H_{ij}(x)$ given by (2.4). We first apply the proximal mapping of F which is given as pointwise geodesic averages of data f_{ij} and the argument of the functional x_{ij}; see (2.11). Then we successively apply the proximal mappings of all the G_{ij}. They are given by (2.6) and (2.8), which is again based on geodesic averaging. As a last step, the analogous operations are executed for the H_{ij}.

Iteration of all these steps yields the algorithm which is stated as Algorithm 1. During the iteration, the parameter λ_r of the proximal mappings is successively decreased. In this way, the penalty for deviation from the previous iterate is successively increased. It is chosen in a way such that the sequence λ_r is square-summable but not summable. This is moderate enough not to prevent convergence towards a minimizer; cf. Theorem 2.

2.3. A parallel proximal point algorithm for TV minimization for manifold-valued data.

Parallel proximal point algorithms were, for example, considered in [16, 29, 76]. We here apply a related parallel proximal point algorithm to TV minimization for manifold-valued data. A great advantage of this approach is its immediate parallelizability.

As for the cyclic algorithm, we split the TV functional into a sum of simpler functionals. But instead of applying the proximal mappings successively, we apply all the proximal mappings to the same initial data. Then the results of the different proximal mappings are averaged.

Algorithm 1: Cyclic proximal point algorithm for ℓ^p-TV^q for manifold data.

```
Input: Manifold-valued image $f \in M^{n\times m}$, regularization parameter $\alpha > 0$, parameter
       sequence for the proximal mappings $\lambda = (\lambda_1, \dots) \in \ell^2 \setminus \ell^1$.
Output: Minimizer $x$ of the $\ell^p$-TV$^q$ problem (2.1).
begin
    $x \leftarrow f$;
    for $r \leftarrow 1, 2, \dots$ do
        for $i \leftarrow 1,\dots,n$; $j \leftarrow 1,\dots,m$ do
            $t \leftarrow \mathrm{calc\_t}_F(\lambda_r, p, q, x_{ij}, f_{ij})$ ;      /* Calculate $t$; see Table 2.  */
            $x_{ij} \leftarrow [x_{ij}, f_{ij}]_t$ ;                                   /* Proximal mapping of $F$. */
        end
        for $i \leftarrow 1,\dots,n$; $j \leftarrow 1,\dots,m-1$ do
            $t \leftarrow \mathrm{calc\_t}_{GH}(\lambda_r\alpha, p, q, x_{ij}, x_{i,j+1})$ ;   /* Calculate $t$; see Table 1.  */
            $x'_{ij} \leftarrow [x_{ij}, x_{i,j+1}]_t$ ;                              /* Proximal mapping of $G_{ij}$. */
            $x'_{i,j+1} \leftarrow [x_{i,j+1}, x_{ij}]_t$;
            $x_{ij} \leftarrow x'_{ij}$; $x_{i,j+1} \leftarrow x'_{i,j+1}$;
        end
        for $i \leftarrow 1,\dots,n-1$; $j \leftarrow 1,\dots,m$ do
            $t \leftarrow \mathrm{calc\_t}_{GH}(\lambda_r\alpha, p, q, x_{ij}, x_{i,j+1})$ ;   /* Calculate $t$; see Table 1.  */
            $x'_{ij} \leftarrow [x_{ij}, x_{i+1,j}]_t$;                               /* Proximal mapping of $H_{ij}$. */
            $x'_{i+1,j} \leftarrow [x_{i+1,j}, x_{ij}]_t$;
            $x_{ij} \leftarrow x'_{ij}$; $x_{i+1,j} \leftarrow x'_{i+1,j}$;
        end
    end
end
```

In order to split the TV functional, we consider the mappings

$$G_e = \sum_{j:j \text{ even}} \sum_i \frac{1}{q} d^q(x_{ij}, x_{i,j+1}),$$

$$G_o = \sum_{j:j \text{ odd}} \sum_i \frac{1}{q} d^q(x_{ij}, x_{i,j+1}). \tag{2.16}$$

The mappings H_e, H_o are defined analogously, exchanging the roles of i and j. Then we have that the ℓ^p-TV^q functional can be decomposed into $F + G_e + G_o + H_e + H_o$. Since $G_e = \sum_{j:j \text{ even}} \sum_i G_{ij}$, the proximal mapping of G_e is explicitly given by

$$(\mathrm{prox}_{\lambda G_e} x)_{i,j} = \begin{cases} [x_{ij}, x_{i,j+1}]_{t_1}, & j \text{ even}, \\ [x_{ij}, x_{i,j-1}]_{t_2}, & j \text{ odd}. \end{cases} \tag{2.17}$$

Here t_1 and t_2 are defined by (2.9), (2.10) (cf. the derivation of (2.8)). The proximal mapping of G_o is obtained by exchanging the terms "even" and "odd" in the above formula. For H_e, H_o, one exchanges the roles of i and j.

Equipped with these explicit formulas for the proximal mappings, the next step is to average the results of the application of the proximal mappings. Since our data live in a

Riemannian manifold, the usual arithmetic mean in a vector space is not available. However, it is well known (cf. [54, 55, 69, 39]) that

$$z^* = \operatorname*{argmin}_{z \in M} \sum_{i=1}^{N} d(z, z_i)^2 \tag{2.18}$$

is the appropriate definition of the mean $z^* = \mathrm{mean}(z_1, \dots, z_N)$ of the N elements z_i on the manifold M. The mean is in general not globally defined since the minimization problem has no unique solution in general. For the z_i being in a small ball, however, it is unique. The size of the ball depends on the sectional curvature of the manifold M. Details and further information can be found, e.g., in [55, 54].

In order to get the mean in the product manifold $M^{n \times m}$ we need only compute the componentwise means. Applied to the above proximal mapping, we get one iteration of the parallel algorithm at pixel i, j by

$$\begin{aligned} &x_{ij}^{(k+1)} \\ &= \mathrm{mean}([x_{i,j}^{(k)}, x_{i,j+1}^{(k)}]_{t_1}, [x_{i,j}^{(k)}, x_{i,j-1}^{(k)}]_{t_2}, [x_{i,j}^{(k)}, x_{i+1,j}^{(k)}]_{t_3}, [x_{i,j}^{(k)}, x_{i-1,j}^{(k)}]_{t_4}, [x_{i,j}^{(k)}, f_{i,j}]_{t_5}), \end{aligned} \tag{2.19}$$

where the t_i are computed according to (2.12), (2.13) and (2.9), (2.10), respectively. So the iterate at pixel (i, j) is obtained by the mean of geodesic averages of the old iterate in a neighborhood. The whole algorithm is summed up as Algorithm 2.

In contrast to the Euclidean case there is no closed form expression of the intrinsic mean defined by (2.18) in Riemannian manifolds. The methods used for computing the mean are of iterative nature and are thus more time consuming. Perhaps the most well known method for computing the intrinsic mean is the gradient descent already mentioned in Karcher's seminal paper [54]; cf. also [39], for example. The iteration for computing the intrinsic mean of the points $x_1, \dots, x_N$ is given by

$$x^{(k+1)} = \exp_{x^{(k)}} \sum_{i=1}^{N} \tfrac{1}{N} \exp_{x^{(k)}}^{-1} x_i. \tag{2.20}$$

Also approaches based on Newton's method can be found in the literature; see, e.g., [36].

However, it is reported in the literature and also confirmed by the authors' experience that the gradient descent converges rather fast; in most cases, 5–10 iterations are enough for five points. This might explain why this simple method of gradient descent is widely used.

The update pattern of our algorithms is related to classical methods for the linear case ($p = q = 2$) and scalar data as follows. For simplicity, we consider the univariate case, i.e., $m = 1$. Then the minimization problem (1.1) reduces to a linear system of the form $(\alpha D^T D + I)x = f$, where D is the first order finite difference matrix and I the identity matrix. The elementary steps in the cyclic algorithm (Algorithm 1) are simultaneous updates in the x_i and the x_{i+1} components. The cyclic algorithm uses the result of the ith update step immediately for the $(i+1)$st step, and thus it can be seen as a Gauss–Seidel-type update scheme. In contrast, Algorithm 2 performs the elementary update steps in parallel, which can be seen as an update pattern of Jacobi type. There is also some similarity to the update pattern of coordinate descent when replacing the canonical vectors e_i by the tuples (e_i, e_{i+1}).

Algorithm 2: Parallel proximal point algorithm for ℓ^p-TV^q for manifold data.

```
Input: Manifold-valued image $f \in M^{n \times m}$, regularization parameter $\alpha > 0$, parameter
       sequence for the proximal mappings $\lambda = (\lambda_1, \dots) \in \ell^2 \setminus \ell^1$.
Output: Minimizer $x$ of the $\ell^p$-TV$^q$ problem (2.1).
begin
    $x \leftarrow f$;
    for $r \leftarrow 1, 2, \dots$ do
        for $i \leftarrow 1, \dots, n; j \leftarrow 1, \dots, m$ do
            $t \leftarrow \mathrm{calc\_t}_F(\lambda_r, p, q, x_{i,j}, f_{i,j})$ ;                /* Calculate $t$; see Table 2.  */
            $z^{(1)} \leftarrow [x_{ij}, f_{ij}]_t$ ;                                            /* Proximal mapping of $F$. */
            $t \leftarrow \mathrm{calc\_t}_{GH}(\lambda_r \alpha, p, q, x_{i,j}, x_{i,j+1})$ ;   /* Calculate $t$; see Table 1.  */
            $z^{(2)} \leftarrow [x_{ij}, x_{i,j+1}]_t$ ;                                         /* Proximal mapping of $G_e/G_o$. */
            $t \leftarrow \mathrm{calc\_t}_{GH}(\lambda_r \alpha, p, q, x_{i,j}, x_{i,j-1})$ ;   /* Calculate $t$ by (2.9),(2.10) */
            $z^{(3)} \leftarrow [x_{ij}, x_{i,j-1}]_t$ ;                                         /* Proximal mapping of $G_o/G_e$. */
            $t \leftarrow \mathrm{calc\_t}_{GH}(\lambda_r \alpha, p, q, x_{i,j}, x_{i+1,j})$ ;   /* Calculate $t$; see Table 1.  */
            $z^{(4)} \leftarrow [x_{ij}, x_{i+1,j}]_t$ ;                                         /* Proximal mapping of $H_o/H_e$. */
            $t \leftarrow \mathrm{calc\_t}_{GH}(\lambda_r \alpha, p, q, x_{i,j}, x_{i-1,j})$ ;   /* Calculate $t$; see Table 1.  */
            $z^{(5)} \leftarrow [x_{ij}, x_{i-1,j}]_t$ ;                                         /* Proximal mapping of $H_o/H_e$. */
            $x'_{ij} \leftarrow \mathrm{mean}(z^{(1)}, z^{(2)}, z^{(3)}, z^{(4)}, z^{(5)})$ ;    /* Intrinsic mean.  */
            Alternative: $x'_{ij} \leftarrow \mathrm{approx\_mean}(z^{(1)}, z^{(2)}, z^{(3)}, z^{(4)}, z^{(5)})$ ;   /* Fast
            approximative variant (cf. (2.24)).  */
        end
        for $i \leftarrow 1, \dots, n; j \leftarrow 1, \dots, m$ do
            $x_{ij} \leftarrow x'_{ij}$;
        end
    end
end
```

2.4. Speedup of the parallel proximal point algorithm.

In Algorithm 2, we calculate the intrinsic mean of the five points $z^{(i)}$ in the inner loop. Each step of the gradient descent (2.20) in the computation of the mean of the $z^{(i)}$ takes about half of the time needed for computing the points $z^{(i)}$ by geodesic averaging. Since we typically need at least five iterations for the gradient descent, computing the means is the most time consuming part of Algorithm 2.

In order to reduce the computation time, we propose replacing the mean by another construction (known as geodesic analogues in the subdivision context [85]) which is computationally less demanding. This construction approximates the mean, the results are comparable (cf. Figure 2), and we can also show convergence towards a minimizer (cf. Theorem 3).

In order to explain the construction, we rewrite the Euclidean mean x of n points $x_1, \dots, x_n$ as iterative convex combinations of only two points (in a binary tree like fashion):

$$x = \sum \frac{1}{n} x_i = \mathrm{conv}_{t_1}(\mathrm{conv}_{t_2}(\dots, \dots \mathrm{conv}_{t_l}(x_{i_l}, x_{j_l}) \dots), \mathrm{conv}_{t_3}(\dots, \dots)). \tag{2.21}$$

Here, we use the notation $\mathrm{conv}(y, z)_t$ for the convex combination $(1-t)y + tz$ of points y, z. For example, for $n = 5$ points, we have the following representation:

$$x = \mathrm{conv}_{0.2}(\mathrm{conv}_{0.5}(\mathrm{conv}_{0.5}(x_1, x_2), \mathrm{conv}_{0.5}(x_3, x_4)), x_5). \tag{2.22}$$

Table 1

Geodesic path length for proximal point problems associated to G and H for the regularization terms considered in this paper.

Regularizer	Geodesic path length (value of $\mathrm{calc\_t}_{GH}(\lambda, p, q, y, z)$)
TV	$\min(\lambda, d(y,z)/2)$
TV^2	$\frac{\lambda}{1+2\lambda} d(y,z)$
Huber	$\begin{cases} \frac{2\lambda\tau^2}{1+4\lambda\tau^2} d(y,z) & \text{if } d(y,z) < \frac{\omega(1+4\lambda\tau^2)}{\sqrt{2}\tau}, \\ \min\left(d(y,z)/2, \sqrt{2}\lambda\omega\tau\right) & \text{otherwise.} \end{cases}$

We note that the above representation is not unique. The idea is now to replace each Euclidean convex combination $\mathrm{conv}_t(x,y)$ in (2.21) by the corresponding Riemannian one, i.e., the point $[x,y]_{td(x,y)}$ on the geodesic joining x and y. Then (2.21) reads as

$$x = [[\ldots, \ldots [x_{i_l}, x_{j_l}]_{t'_l} \ldots]_{t'_2}, [\ldots, \ldots]_{t'_3}]_{t'_1}, \tag{2.23}$$

where $t'_k = t_k d_k$ and d_k denotes the distance of the elements in the bracket to which the d_k is attached. (This technicality arises since we consider unit speed geodesics.) We consider the above decomposition (2.22) and transport it to the Riemannian setting (2.23). Then, instead of using the mean in Algorithm 2 we propose using the alternative procedure (called "approx_mean" in Algorithm 2) given by

$$x = [[[z^{(1)}, z^{(2)}]_{0.5d_1}, [z^{(3)}, z^{(4)}]_{0.5d_2}]_{0.5d_3}, z^{(5)}]_{0.2d_4}. \tag{2.24}$$

Here each d_k again denotes the distance of the elements in the bracket to which the d_i is attached. The points $z^{(i)}$ are the results of the application of the proximal mappings in Algorithm 2.

The full algorithm is given by Algorithm 2 using the part referred to as "Alternative."

2.5. Huber regularizing and data terms.

TV regularized images may suffer from the undesirable creation of steps in the solution. This is often called staircasing effect. An effective way to decrease staircasing is to replace the total variation by the *Huber regularizer*, sometimes called the *Huber-ROF model* [21]. To this end, we replace d^q in the definition of the TV^q functional (2.1) by $h \circ d$, which is the concatenation of the distance on the manifold and the *Huber function* h. The Huber function h is defined, for $s > 0$, by

$$h(s) = \begin{cases} \tau^2 s^2 & \text{for } s < \omega/(\sqrt{2}\tau), \\ \omega\sqrt{2}\tau s - \omega^2/2 & \text{otherwise,} \end{cases} \qquad \tau, \omega > 0. \tag{2.25}$$

It is a square function (for small arguments) smoothly glued with an absolute value function (for large arguments). In analogy to (2.3) and (2.4) we write the Huber regularizer as $\sum_{i,j} G^h_{ij} + \sum_{i,j} H^h_{ij}$ with $G^h_{ij}(x) = h \circ d(x_{ij}, x_{i,j+1})$ and $H^h_{ij}(x) = h \circ d(x_{ij}, x_{i+1,j})$. As in (2.6) and (2.8), the proximal mappings of the G^h_{ij} are given by (cf. the proof of Theorem 2)

$$(\mathrm{prox}_{\lambda G^h_{ij}} x)_{kl} = \begin{cases} [x_{ij}, x_{i,j+1}]_t, & k = i \text{ and } l = j, \\ [x_{i,j+1}, x_{ij}]_t, & k = i \text{ and } l = j+1, \\ x_{kl} & \text{otherwise,} \end{cases} \tag{2.26}$$

Table 2

Geodesic path length for the proximal point problem associated to F for the data terms considered in this paper.

Data term	Geodesic path length (value of calc_t$_F(\lambda, p, q, y, z)$)
ℓ^1	$\min(\lambda, d(y,z))$
ℓ^2	$\frac{\lambda}{1+\lambda} d(y,z)$
Huber	$\begin{cases} \frac{2\lambda\tau^2}{1+2\lambda\tau^2} d(y,z) & \text{if } d(y,z) < \frac{\omega(1+2\lambda\tau^2)}{\sqrt{2}\tau}, \\ \min\left(d(y,z), \sqrt{2}\lambda\omega\tau\right) & \text{otherwise.} \end{cases}$

where

$$t = \begin{cases} \frac{2\lambda\tau^2}{1+4\lambda\tau^2} d(x_{i,j}, x_{i,j+1}) & \text{if } d(x_{ij}, x_{i,j+1}) < \frac{\omega(1+4\lambda\tau^2)}{\sqrt{2}\tau}, \\ \min\left(d(x_{ij}, x_{i,j+1})/2, \sqrt{2}\lambda\omega\tau\right) & \text{otherwise.} \end{cases} \tag{2.27}$$

So replacing the procedure to calculate the geodesic length t in Algorithms 1 and 2 by the procedure to calculate t given by (2.27) yields cyclic and parallel minimization algorithms with the Huber regularizing term.

We can use the Huber function for the data term; i.e., we let $F_h(x,f) = \sum_{i,j} h \circ d(x_{ij}, f_{ij})$. For small distances the Huber data term behaves like the ℓ^2 data term, but it is more robust to outliers. The proximal mapping of F_h is given by (cf. the proof of Theorem 2)

$$\begin{aligned} &(\mathrm{prox}_{\lambda F_h})_{ij}(x) = [x_{ij}, f_{ij}]_t, \\ &\quad \text{where } t = \begin{cases} \frac{2\lambda\tau^2}{1+2\lambda\tau^2} d(x_{ij}, f_{ij}) & \text{if } d(x_{ij}, f_{ij}) < \frac{\omega(1+2\lambda\tau^2)}{\sqrt{2}\tau}, \\ \min\left(d(x_{ij}, f_{ij}), \sqrt{2}\lambda\omega\tau\right) & \text{otherwise.} \end{cases} \end{aligned} \tag{2.28}$$

Using the proximal mapping of the Huber data term (2.28) instead of the proximal mapping of F in Algorithms 1 and 2, respectively, yields cyclic and parallel TV minimization algorithms with the Huber data term.

3. Convergence in Hadamard spaces.

In this section, we show the convergence of Algorithms 1 and 2 to a global minimizer for a certain class of spaces on which the TV functionals we consider are convex.

For general Riemannian manifolds, the ℓ^p-TV^q functional (2.1) is not necessarily convex. The perhaps simplest example where convexity fails is the one-dimensional (1D) sphere S^1 (cf. [78]). In the nonconvex case, the study of (global) convergence becomes much more involved and is out of the scope of this paper. Here we treat the quite large class of Cartan–Hadamard manifolds where we still have convexity. These are complete Riemannian manifolds of nonpositive sectional curvature. Prominent examples are the spaces of positive matrices (which are the data space in diffusion tensor imaging) and the hyperbolic spaces. For details we refer the reader to [32] or to [10].

The proofs in this section work in the more general setup of Hadamard spaces without additional effort. Hadamard spaces are certain metric spaces generalizing the concept of Cartan–Hadamard manifolds. Examples of Hadamard spaces which are not Cartan–Hadamard manifolds are the metric trees of [80]. Since there is no additional effort, we consider Hadamard spaces as underlying spaces in this section.

A Hadamard space is a geodesic space, i.e., for each two points x, y there is an arc connecting them and the length of the shortest such arc connecting x and y equals the distance of the points; see, e.g., [80]. Furthermore, there is a certain condition ensuring that the geodesic triangles are "not fat." (Triangles on the sphere are "fat.") This condition providing the essential properties of Cartan–Hadamard manifolds in this metric space setting is as follows. For given x_0, x_1 there is a point y on the geodesic joining them such that for every z, $d^2(z,y) \leq \frac{1}{2}d^2(z,x_0) + \frac{1}{2}d^2(z,x_1) - \frac{1}{4}d^2(x_0,x_1)$. For details on Hadamard spaces we refer the reader to [80] and the references therein or to the book [17].

We note that the ℓ^p-TV^q functionals given by (2.1) are convex in a Hadamard space. This is because the distance function is doubly convex in Hadamard spaces; see [80].

We formulate our convergence results for Algorithms 1 and 2 in section 3.1. The proofs are derived in section 3.2.

3.1. Formulation of the main results on convergence.

Our first result is the convergence of Algorithm 1, which is the geodesic averaging algorithm based on cyclical application of proximal mappings.

Theorem 2. *For data in a (locally compact) Hadamard space, Algorithm 1 converges towards a minimizer of the ℓ^p-TV^q functional. The statement remains true when using Huber data and regularizing terms based on* (2.25).

Furthermore, we obtain the convergence of the parallel proximal point algorithm, Algorithm 2, and its fast variant (with the approximate mean calculation from (2.23)) in a Hadamard space.

Theorem 3. *The parallel proximal algorithm for ℓ^p-TV^q minimization (Algorithm 2) and its approximative variant converge towards a minimizer in every (locally compact) Hadamard space. The statement remains true when using Huber regularization and Huber data terms based on* (2.25).

3.2. Proof of the main results on convergence.

Our first goal is to show Theorem 2, which ensures the convergence of Algorithm 1. This is the geodesic averaging algorithm based on the cyclical application of proximal mappings.

Proof of Theorem 2. We first show that, in a Hadamard space, the proximal mappings of the functions F, G_{ij}, and H_{ij} are given by (2.11), (2.6), and (2.8) and their analogues for H_{ij}. We also show that the proximal mappings of the Huber regularizing and data terms are given by (2.26), (2.27), and (2.28), respectively. We start with the mappings G_{ij}. The proximal mapping of G_{ij} is given by

$$\mathrm{prox}_{\lambda G_{ij}} x = \operatorname*{argmin}_{y} \lambda \tfrac{1}{q} d(y_{ij}, y_{i+1,j})^q + \tfrac{1}{2} \sum_{k,l} d(y_{kl}, x_{kl})^2. \tag{3.1}$$

Hence, every minimizer y^* must fulfill $y^*_{kl} = x_{kl}$ for $k \neq i$ and $l \neq j, j+1$. Otherwise, letting $y^*_{kl} = x_{kl}$ for $k \neq i$ and $l \neq j, j+1$ would decrease the functional value, which contradicts the minimizer property. This implies (2.6).

Now let y^* be a minimizer of (3.1). We show that the four points $x_{ij}, x_{i,j+1}, y^*_{ij}$, $y^*_{i,j+1}$ lie on one geodesic. We may assume that $d(x_{ij}, y^*_{ij}) \leq d(x_{ij}, x_{i,j+1})$ since, otherwise, setting

$y'_{ij} = y'_{i,j+1} = x_{i,j+1}$ would yield a lower functional value in (3.1). By the same argument, we may assume that $d(x_{i,j+1}, y^*_{i,j+1}) \leq d(x_{ij}, x_{i,j+1})$.

We define the point $z = [x_{ij}, x_{i,j+1}]_{t_1}$ as the point reached on the unit speed geodesic starting at x_{ij} after time $t_1 = d(x_{ij}, y^*_{ij})$. Analogously, we let $z' = [x_{i,j+1},\ x_{i,j+1}]_{t_2}$ be the point on the same geodesic when starting from x_{k+1} after time $t = d(x_{k+1}, y^*_k)$.

We first consider the case with ordering $x_{ij}, z, z', x_{i,j+1}$ when running on the geodesic starting at x_k (including the case $z = z'$.) Since these points lie on a geodesic, we have that

$$\begin{aligned} d(x_{ij}, x_{i,j+1}) &= d(x_{ij}, z) + d(z, z') + d(z', x_{i,j+1}) \\ &\leq d(x_{ij}, y^*_{ij}) + d(y^*_{ij}, y^*_{i,j+1}) + d(y^*_{i,j+1}, x_{i,j+1}). \end{aligned} \tag{3.2}$$

Here the inequality is true since every geodesic is a shortest path in a Hadamard space. By our choice of t_1, t_2, this implies $d(z, z') \leq d(y^*_{ij}, y^*_{i,j+1})$. As a consequence, the functional value $a(z, z') \leq a(y^*_{ij}, y^*_{i,j+1})$, where

$$a(v, v') = \tfrac{1}{2} d(v, x_{ij})^2 + \lambda \tfrac{1}{q} d(v, v')^q + \tfrac{1}{2} d(v', x_{i,j+1})^2. \tag{3.3}$$

This is the essential part of the functional in (3.1), meaning that minimizing the functional in (3.1) is equivalent to minimizing a. Hence, since geodesics are unique in a Hadamard space, $z = y^*_{ij}$ and $z' = y^*_{i,j+1}$. Thus, these four points lie on a geodesic.

Next, we consider the case with ordering x_{ij}, z', z, $x_{i,j+1}$ when running on the geodesic starting at x_k. Then we have that $a(z, z) \leq a(z, z')$ since $d(z, x_{ij}) < d(z, x_{i,j+1})$. Hence, we obtain a lower functional value, which means that this situation cannot occur for a minimizer of (3.1).

Summing up, we know that the points $x_{ij}, y^*_{ij}, y^*_{i,j+1}, x_{i,j+1}$ lie on a geodesic in this ordering.

Next, we need the precise position of $y^*_{ij}, y^*_{i,j+1}$ on the geodesic. We consider the real numbers $d = d(x_{ij}, x_{i,j+1})$ and the time points t_1 and t_2 given above. Minimization of a is now equivalent to minimizing, for $0 \leq t_1, t_2 \leq d/2$,

$$a'(t_1, t_2) = \tfrac{1}{2} t_1^2 + \lambda \tfrac{1}{q} (d - t_1 - t_2)^q + \tfrac{1}{2} t_2^2.$$

By symmetry and uniqueness, a minimizer fulfills $t_1 = t_2$. Hence, we have to find a minimizer of

$$a''(t) = t^2 + \tfrac{\lambda}{q} (d - 2t)^q.$$

For $q = 1$, we get the solution

$$t = \min(\lambda, d/2),$$

and for $q = 2$,

$$t = \frac{\lambda d}{2 + 2\lambda}.$$

This implies (2.8) and the subsequent formulas (2.9) and (2.10). The corresponding proof for the H_{ij} is analogous.

Next, we consider the Huber data term F_h based on (2.25). We show that its proximal mapping is given by (2.28). Similarly as above, we define the point $z = [x_{ij}, f_{ij}]_t$, where

$t = d(x_{ij}, y^*_{ij})$ and y^*_{ij} is the (i, j)th component of the proximal mapping of λF_h at x. Modifying the arguments above, we see that $z = y^*_{ij}$. Thus the three points x_{ij}, y^*_{ij}, f_{ij} lie on one geodesic and it remains to determine t. This leads to minimizing the scalar problem $t \mapsto 2\lambda h(d-t)+t^2$ under the constraint $0 \leq t \leq d$, where we let $d = d(x_{ij}, f_{ij})$. To solve this problem one applies a calculation analogous to that in Example 4.5 of [24] and concludes (2.28). The corresponding proof of (2.11) for the ℓ^p-type data term F is analogous.

It remains to consider the Huber regularizer $G^h + H^h$ from section 2.5. We proceed analogous to the proof for the regularizer $G + H$ to obtain that the four points $x_{ij}, y^*_{ij}, y^*_{i,j+1}, x_{i,j+1}$ lie on a geodesic in this ordering which shows (2.26). Then proceeding as in the proof for the Huber data term F_h with a similar calculation as in [24] we obtain the formula (2.27).

Since Algorithm 1 produces only convex combinations of the points involved, we have that the iterates produced by the algorithm stay in the convex hull of the data $(f_{i,j})_{i,j}$. Since all functions F, G_{ij}, H_{ij} are continuous, they are Lipschitz on that convex hull. Hence the assumptions of [9, Thm. 3.4] are fulfilled, and the application of this theorem yields the convergence of Algorithm 1. ■

The goal of the rest of this section is to show that Algorithm 2 and its fast variant (with the approximate mean calculation from (2.23)) converge in a Hadamard space. To this end, we first show a generic convergence statement for parallel proximal point algorithms.

Theorem 4. *We consider a convex function g defined on a Hadamard space which has a minimizer. Let $g = g_1 + \cdots + g_n$, and assume that all summands are convex and lower semicontinuous. Assume further that the positive parameter sequence $\lambda = (\lambda_1, \ldots)$ is square-summable but not summable. We consider the iteration*

$$x^{k+1} = \text{mean}\left(\text{prox}_{\lambda_k g_1} x^k, \ldots, \text{prox}_{\lambda_k g_n} x^k\right). \tag{3.4}$$

Here mean *is the intrinsic mean in the Hadamard space defined by* (2.18). *If there is a constant $L > 0$ such that, for all g_i and all k,*

$$g_i(x^k) - g_i(\text{prox}_{\lambda_k g_i} x^k) \leq L \cdot d(x^k, \text{prox}_{\lambda_k g_i} x^k), \tag{3.5}$$

then the iteration (3.4) *converges to a minimizer of g.*

The proof of this statement is an adaption of the proof of Theorem 3.4 in [9] to the parallel setting. In [9], the applied method of proof is addressed to [13]. We need the following two lemmas, which are Lemmas 2.6 and 3.2 of [9].

Lemma 5. *Let a_k, b_k, c_k be sequences of positive numbers. Assume that $\sum c_k < \infty$ and that, for all k, $a_{k+1} < a_k - b_k + c_k$. Then the sequence a_k converges and $\sum b_k < \infty$.*

Lemma 6. *Consider a convex and lower semicontinuous function h on a (locally compact) Hadamard space. Then*

$$h(\text{prox}_{\lambda h} x) - h(y) \leq \tfrac{1}{2\lambda}\left(d(x,y)^2 - d(\text{prox}_{\lambda h} x, y)^2\right) \tag{3.6}$$

for any y in the Hadamard space.

Related results in the context of Riemannian manifolds (and not Hadamard spaces) can be found in [4, 5]. Equipped with these preparations, we show Theorem 4.

Proof of Theorem 4. The function $x \mapsto d(x,y)^2$ is uniformly convex (cf. [80]). Thus, using Jensen's inequality (cf. [80]), we get, for the intrinsic mean in the Hadamard space x^{k+1}, that

$$d(x^{k+1},y)^2 \leq \sum_{i=1}^{n} \tfrac{1}{n} d(\operatorname{prox}_{\lambda_k g_i} x^k, y)^2 \tag{3.7}$$

for all y. In the following, we use the notation $\tilde{x}_i^{k+1} = \operatorname{prox}_{\lambda_k g_i} x^k$ for the proximal mapping of g_i at the previous iterate x^k. Using Lemma 6, we estimate

$$\begin{aligned} d(\tilde{x}_i^{k+1},y)^2 &\leq d(x^k,y)^2 - 2\lambda_k \Big(g_i(\tilde{x}_i^{k+1}) - g_i(y) \Big) \\ &= d(x^k,y)^2 - 2\lambda_k \Big(g_i(x^k) - g_i(y) \Big) + 2\lambda_k \Big(g_i(x^k) - g_i(\tilde{x}_i^{k+1}) \Big). \end{aligned} \tag{3.8}$$

We combine the estimates (3.7) and (3.8) to obtain

$$\begin{aligned} &d(x^{k+1},y)^2 \\ &\quad \leq \frac{1}{n}\sum_{i=1}^{n} d(x^k,y)^2 - \frac{2\lambda_k}{n}\sum_{i=1}^{n} (g_i(x_n) - g_i(y)) + \frac{2\lambda_k}{n}\sum_{i=1}^{n} \Big(g_i(x_n) - g_i(\tilde{x}_i^{k+1}) \Big) \\ &\quad = d(x^k,y)^2 - \frac{2\lambda_k}{n}(g(x_n) - g(y)) + \frac{2\lambda_k}{n}\sum_{i=1}^{n} \Big(g_i(x_n) - g_i(\tilde{x}_i^{k+1}) \Big). \end{aligned} \tag{3.9}$$

The next goal is to estimate the last summand on the right-hand side. To this end, we use that, by assumption,

$$g_i(x^k) - g_i(\tilde{x}_i^{k+1}) \leq L d(x^k, \tilde{x}_i^{k+1}).$$

Furthermore, since $\tilde{x}_i^{k+1}$ minimizes the expression in the definition of the proximal mapping, we obtain

$$g_i(\tilde{x}_i^{k+1}) + \tfrac{1}{2\lambda_k} d(x^k, \tilde{x}_i^{k+1}) \leq g_i(x^k).$$

Applying these estimates successively yields

$$\begin{aligned} g_i(x^k) - g_i(\tilde{x}_i^{k+1}) &\leq L d(x^k, \tilde{x}_i^{k+1}) \\ &\leq L 2\lambda_k \frac{g_i(x^k) - g_i(\tilde{x}_i^{k+1})}{d(x^k, \tilde{x}_i^{k+1})} \leq 2\lambda_k L^2. \end{aligned} \tag{3.10}$$

This allows us to estimate the last summand on the right-hand side of (3.9) by

$$\frac{2\lambda_k}{n}\sum_{i=1}^{n} \Big(g_i(x_n) - g_i(\tilde{x}_i^{k+1}) \Big) \leq \frac{2\lambda_k}{n}\sum_{i=1}^{n} 2\lambda_k L^2 = 4\lambda_k^2 L^2.$$

Thus, (3.9) now reads as

$$d(x^{k+1},y)^2 \leq d(x^k,y)^2 - \frac{2\lambda_k}{n}(g(x_n) - g(y)) + 4\lambda_k^2 L^2. \tag{3.11}$$

Next, we consider a minimizer y^* of g and plug it into (3.11) above. Then $g(x_n)-g(y^*) \geq 0$, and we may apply Lemma 5 to (3.11). This yields that $d(x^k, y^*)$ converges as $n \to \infty$ and that

$$\sum_{n\in\mathbb{N}} \frac{2\lambda_k}{n} \left(g(x_n) - g(y)\right) < \infty. \tag{3.12}$$

Since the parameter sequence $\lambda = (\lambda_1, \dots)$ is not summable, (3.12) implies that $g(x^{k_l}) \to g(y^*)$ on a subsequence k_l. Since $d(x^k, y^*)$ is bounded and the underlying space is locally compact, we may choose another subsequence k_{l_r} such that x^k converges on this subsequence; call the limit x^*. Then by the lower semicontinuity of g, we have $g(x^*) \leq g(y^*)$. Then, since y^* is a minimizer, also x^* is a minimizer. We apply Lemma 5 for a second time, but now for $y = x^*$. As a result $d(x^k, x^*)$ converges. Moreover, $d(x^k, x^*) \to 0$ since this is true on a subsequence. This completes the proof. ■

For the fast variant of the parallel proximal algorithm introduced in section 2.4 we replaced the intrinsic mean by the approximation (2.23). In order to obtain convergence of the corresponding algorithm, we need the following result.

Theorem 7. *The statement of Theorem 4 remains true if we replace the intrinsic mean (2.18) by its approximation (2.23).*

Proof. We show that (3.7) remains true if we replace the intrinsic mean by the construction given in (2.23). By the convexity of the function $a(z) = d(z, y)^2$ we have that $a([z, y]_{td(z,y)}) \leq (1-t)a(z) + ta(y)$ for all $z, y \in M$. We successively apply this inequality to every geodesic average in (2.23) (in a top-down fashion). As a first step, we obtain

$$a(x) \leq (1-t_1')a([\dots,\dots[x_{i_l}, x_{j_l}]_{t_l'}\dots]_{t_2'}) + t_1' a([\dots,\dots]_{t_3'}).$$

Proceeding further, we get

$$a(x) \leq c_1 a(x_1) + \cdots + c_n a(x_n), \tag{3.13}$$

where the c_k are products with factors t_r' and $(1-t_r')$. Reversing the construction of the t_r' in (2.21), we see that, for each x_k, the factor c_k equals $1/n$. Plugging the definition $a(z) = d(z, y)^2$ and $c_k = 1/n$ in (3.13) yields (3.7) for the construction (2.23). Then we can follow the rest of the proof of Theorem 4 to conclude the assertion of the present theorem. ■

Equipped with these preparations we can now prove our main results concerning the convergence of the parallel proximal algorithm which was formulated as Theorem 3.

Proof of Theorem 3. As a first step, we have to show that the proximal mappings of the functions G_e, G_o, H_e, H_o given in and below (2.16) are in fact given by (2.17) and the explanations following (2.17). We consider only G_e since the other cases are analogous. We use the fact that G_e is a sum of the functions G_{ij}. More precisely, $G_e = \sum_{j:j \text{ even}} \sum_i G_{ij}$. Considering this form of G_e, we see that the minimization problem in the definition of the proximal mapping separates into minimization problems which require minimizing expressions of the form (3.3). Hence, the (i,j)th component of the proximal mapping of G_e equals the corresponding component of the proximal mapping of G_{ij}. This proximal mapping has been considered in the proof of Theorem 2. Its (i,j)th component agrees with the expression in (2.17), which proves (2.17).

For the Huber regularizing term, we consider the functionals $G_e^h, G_o^h, H_e^h, H_o^h$ defined in analogy to G_e, G_h, H_e, H_o by replacing d^q by $h \circ d$, where d is the Huber function (2.25) in (2.16). Then following the argument for G_e in the previous paragraph, we see that the (i, j)th component of the proximal mappings of G_e^h agrees with the corresponding component of the proximal mapping of G_{ij}^h given by (2.26), (2.27). Analogous statements are true for G_o^h, H_e^h, H_o^h. The proximal mappings of the ℓ^p-type data term F and the Huber data term F_h have been shown to agree with (2.11) and (2.28), respectively, in the proof of Theorem 2.

The next step is to apply Theorems 4 and 7 (for the approximative variant). Since the algorithm produces only convex combinations and intrinsic means, the iterates produced by the algorithm stay in the convex hull of the data $(f_{i,j})_{i,j}$. So the involved functions (which are all continuous) are Lipschitz on this convex hull, which means that (3.5) is fulfilled. Hence, we may apply Theorems 4 and 7 and conclude the assertion of the theorem. ■

4. Applications. In this section, we apply the algorithms proposed in this paper to concrete manifolds which frequently occur in applications. The manifolds we consider are the space of positive matrices Pos_3, the spheres S^1 and S^2, and the product space $S^1 \times \mathbb{R}^2$ (which appears in the context of nonlinear color models) as well as the rotation group.

In order to make Algorithms 1 and 2 work in a specific manifold we have to compute geodesics and distances on this manifold. This is accomplished using the Riemannian exponential mapping and its inverse. Recall that the exponential mapping $\exp_a : T_aM \to M$ returns the point $\exp_a v$ on the manifold which we obtain when following the unit speed geodesic starting at a into the direction of the given tangent vector v for time $\|v\|_a$. Conversely, the inverse of the exponential mapping $\exp_a^{-1} : M \to T_aM$ gives us the tangent vector $\exp_a^{-1} b$ at the point a which leads to the point b when following the geodesic with respect to this tangent vector for time $\| \exp_a^{-1} b\|_a$. Using these mappings, the point $[a, b]_t$ reached on the unit speed geodesic joining a and b after time t is given by

$$[a, b]_t = \exp_a(t \cdot \exp_a^{-1}(b)). \tag{4.1}$$

In order to calculate the geodesic path length t, we further have to calculate distances on the manifold under consideration (cf. Tables 1 and 2). To this end, we use that the distance between points a and b is given by the length of the tangent vector $\exp_a^{-1}(b)$, i.e.,

$$d(a, b) = \| \exp_a^{-1}(b)\|_a. \tag{4.2}$$

Here the length is measured with respect to the Riemannian metric in the tangent space of a. Hence, in order to apply our algorithms for a specific data space, we need only instantiate the exponential mapping and its inverse for the corresponding manifold. For the data spaces considered in this paper, the exponential mappings and their inverse have closed expressions involving only basic arithmetic operations such as trigonometric functions or matrix exponentials.

The numerical experiments were conducted on a Macbook using a single core of a 2.6 GHz Intel Core i7 processor. (Parallelized implementations of our algorithms are out of the scope of this paper.) For the experiments in Figure 5, we optimized the TV parameter α with respect to the peak signal-to-noise ratio. In the other experiments, α was determined empirically. A simple choice for the sequence λ_r is $\lambda_r = cr^{-\omega}$ with $c > 0$. The sequence fulfills the condition to be in $\ell^2 \setminus \ell^1$ for each $0.5 < \omega \leq 1$. We here used $\omega = 0.95$ and $c = 3$. We observed only little differences when using different parameter pairs. In order to quantitatively assess the denoising performance of TV regularization on manifolds we use the *signal-to-noise ratio improvement* (cf. [82, Chap. 10]). We consider a manifold-valued version of the signal-to-noise ratio improvement which is given by

$$\Delta\mathrm{SNR} = 10\log_{10}\left(\frac{\sum_{ij} d(g_{ij}, f_{ij})^2}{\sum_{ij} d(g_{ij}, x_{ij})^2}\right).$$

Here f is the noisy data, g the ground truth, and x the regularized restoration.

4.1. The space of positive matrices Pos_3: Diffusion tensor imaging.

Diffusion tensor imaging (DTI) is a noninvasive imaging modality based on nuclear magnetic resonance. It allows one to quantify the diffusional characteristics of a specimen [11, 52]. Applications are the determination of fiber tract orientations [11] and the detection of brain ischemia [59]. Denoising is an important topic in DTI which has been addressed in various articles; see, e.g., [26, 69, 12].

In DTI, the diffusivity of water molecules is captured by a *diffusion tensor*, i.e., a (symmetric) positive (definite) 3×3 matrix $S(p)$ sitting at pixel p. It is reasonable to consider the space of diffusion tensors Pos_3 as a Riemannian manifold with the Riemannian metric

$$g_D(W, V) = \mathrm{trace}(D^{-\frac{1}{2}} W D^{-1} V D^{-\frac{1}{2}});$$

see [69]. Here the symmetric matrices W, V represent tangent vectors in the point D. Equipped with this Riemannian metric the space of positive matrices becomes a Cartan–Hadamard manifold. Hence, by virtue of Theorems 2 and 3, the cyclic proximal point algorithm and both variants of the parallel algorithm converge to a global minimizer.

For the space of positive matrices, the Riemannian exponential mapping $\exp_D$ is given by

$$\exp_D(W) = D^{\frac{1}{2}} \exp(D^{-\frac{1}{2}} W D^{-\frac{1}{2}}) D^{\frac{1}{2}}.$$

Here D is a positive matrix and the symmetric matrix W represents a tangent vector in D. The mapping exp is the matrix exponential. Furthermore, there is also a closed form expression for the inverse of the Riemannian exponential mapping: we have, for positive matrices D, E,

$$\exp_D^{-1}(E) = D^{\frac{1}{2}} \log(D^{-\frac{1}{2}} E D^{-\frac{1}{2}}) D^{\frac{1}{2}}.$$

The matrix logarithm log is well defined since the argument is a positive matrix. The matrix exponential and logarithm can be efficiently computed by diagonalizing the symmetric matrix under consideration and then applying the scalar exponential and logarithm functions to the eigenvalues. The distance between D and E is just the length of the tangent vector $\exp_D^{-1}(E)$ which can be explicitly calculated by

$$d^2(D, E) = \sum_{l=1}^{3} \log(\kappa_l)^2,$$

where κ_l is the lth eigenvalue of the matrix $D^{-\frac{1}{2}}ED^{-\frac{1}{2}}$.

The data actually measured in DTI are so-called diffusion weighted images (DWIs) $D_v(p)$, which capture the directional diffusivity in the direction v at pixel p. The relation between the diffusion tensor $S(p)$ and the DWIs $D_v(p)$ at the pixel p is given by the Stejskal–Tanner equation

$$D_v(p) = A_0 e^{-b\ v^T S(p) v} \tag{4.3}$$

with constants $b, A_0 > 0$. Typically $b = 800$ and $A_0 = 1000$. Usually, 6 to 30 DWIs D_v (with different directions v) are measured [52, sect. 3. IV C]. Being magnetic resonance images the DWIs are corrupted by Rician noise which arises from complex-valued Gaussian noise in the original frequency domain measurements [12]. This means that assuming the model (4.3) the actual measurement in direction v at pixel p is given by

$$D'_v(p) = \sqrt{(X + D_v(p))^2 + Y^2}$$

with the Gaussian random variables $X, Y \sim N(0, \sigma^2)$. Typically, the tensor $S(p)$ is obtained from the DWIs via a least square fit using the Stejskal–Tanner equation (4.3). In our synthetic examples, we impose Rician noise to 15 DWIs D'_v obtained from a synthetic diffusion tensor image S by (4.3). Then we apply least square fitting to the noisy DWIs to obtain a noisy diffusion tensor image.

In our experiments we visualize the diffusion tensors by the isosurfaces of the corresponding quadratic forms. More precisely, the ellipses visualizing the diffusion tensor $S(p)$ at pixel p are the points x fulfilling $(x-p)^T S(p)(x-p) = c$ for some $c > 0$. We work on slices which are 2D domains. However, we respect the 3D nature of DTI since the data we use are 3×3 matrices.

Denoising DTI images using a Rician noise model was also performed in [37]. There a smooth anisotropic regularizer is applied, whereas we use a nonsmooth jump preserving TV term; furthermore, there, a Log-Euclidean metric is used which results in transporting the problem to the tangent space of the identity matrix and solving it in this linear space; see also [6, 7]. Our approach is differential-geometric in the sense that it uses no specific base point; cf. [69]. Approximation by going to a tangent space at a base point b yields good results when the involved points are near b; when this is not the case, one loses approximation quality.

In Figure 1, we apply ℓ^2-TV minimization to real DTI data of a human brain. The data set stems from the Camino project [30] and is freely accessible. We observe that TV minimization removes noise and preserves sharp boundaries between oriented structures.

In Figure 2, we apply Algorithm 1 as well as Algorithm 2 and its fast variant for ℓ^2-TV minimization to a synthetic DTI on which we impose Rician noise. We observe the denoising capabilities of the proposed algorithms under a relatively high noise level; minimization of the ℓ^2-TV functional almost completely removes the noise while preserving sharp boundaries at the same time.

In Figure 3, we compare our result using Algorithm 1 with the result obtained from TV minimization using the Euclidean metric on the cone of positive matrices. For TV minimization in the Euclidean case, we use the algorithm of Chambolle and Pock [21]. In the context of processing DTI data using the Euclidean metric, often a swelling effect is reported

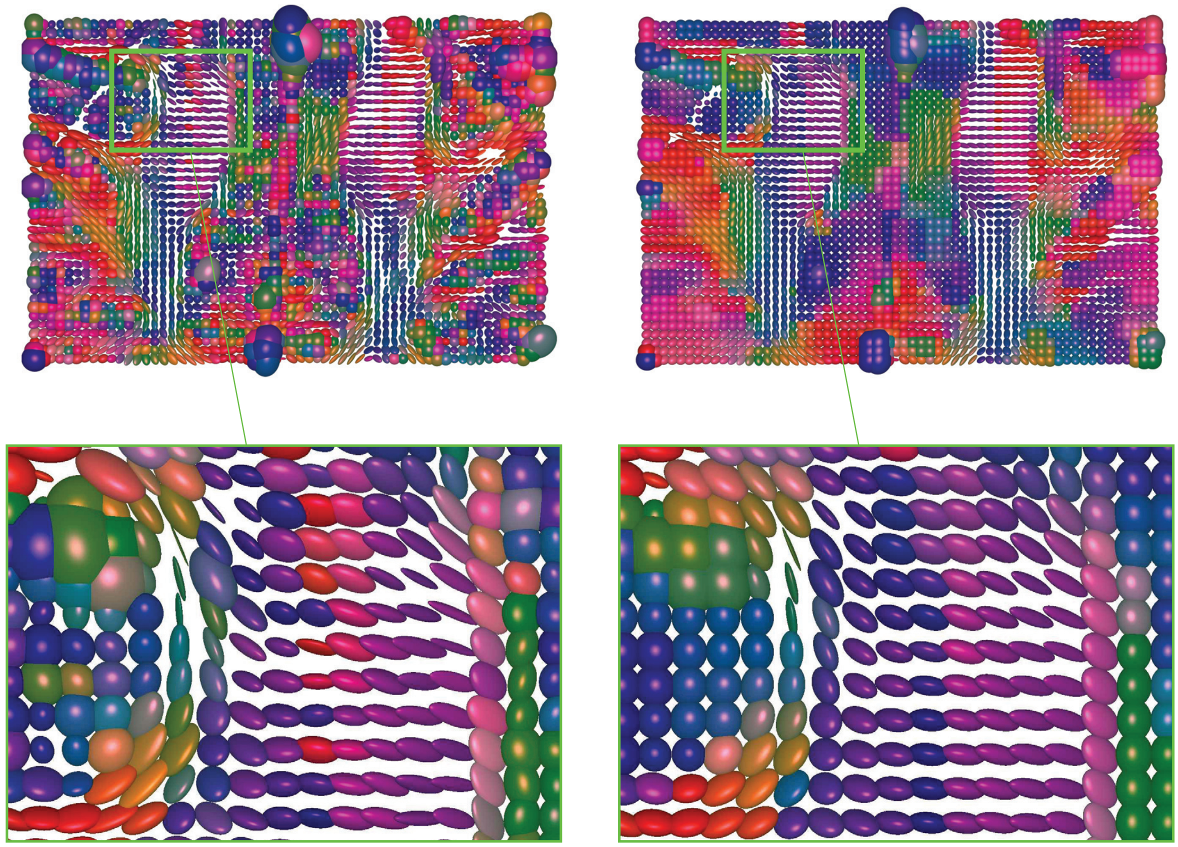
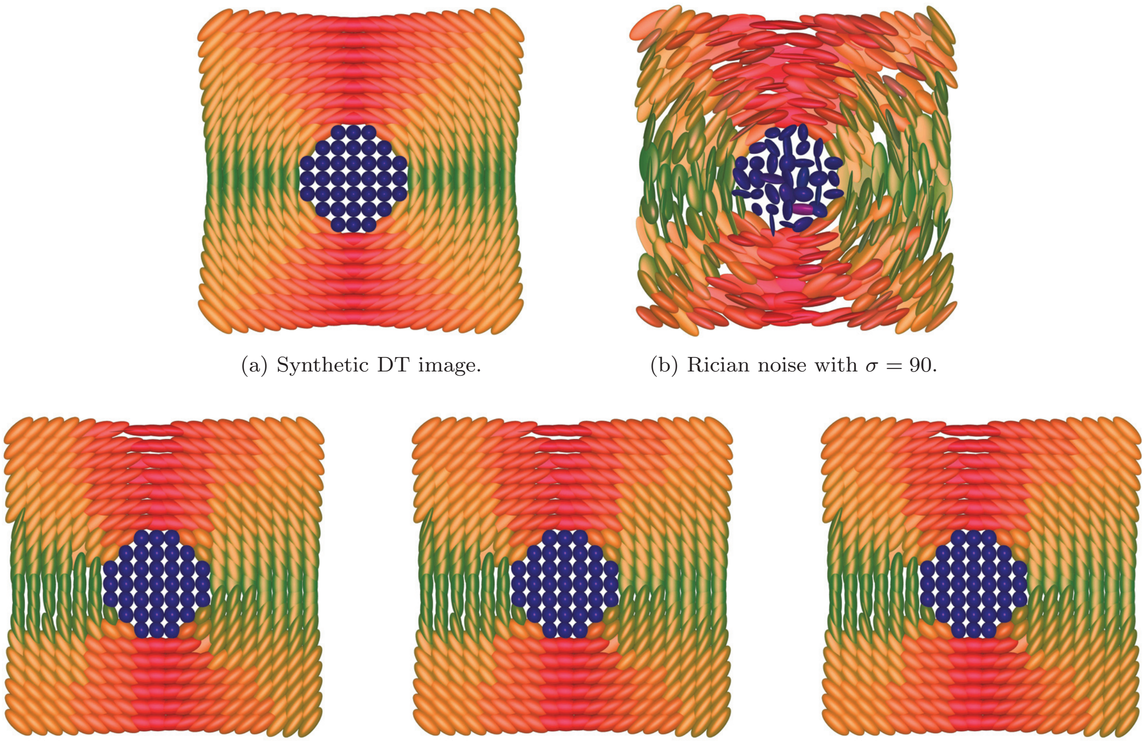


Figure 1. Left: *Diffusion tensor image of a human brain (axial cut);* Right: *TV denoising with the ℓ^2 data term using the cyclic proximal point algorithm (Algorithm 1) using $\alpha = 0.11$. The runtime is 496.0 sec for 4000 iterations. The regularized image is much smoother than the original image. At the same time, strong changes of the orientations are preserved.*

[81, 6]: this means that the determinant of the considered matrices and thus the dispersion of the corresponding covariance matrices tend to be larger than the original ones [6]. In the experiment, we also observe a slight swelling in the Euclidean setup. The swelling effect is less pronounced when using our approach for TV minimization based on the affine-invariant Riemannian metric.

4.2. The 1D sphere S^1: InSAR images. Synthetic aperture radar (SAR) is a radar technique for sensing the earth's surface from measurements taken by aircrafts or satellites. Interferometric synthetic aperture radar (InSAR) images consist of the phase difference between two SAR images, recording a region of interest either from two different angles of view or at two different points in times. Important applications of InSAR are the creation of accurate digital elevation models and the detection of terrain changes; cf. [60, 72].

As InSAR data consist of phase values, the natural data space of InSAR images is the 1D sphere S^1. The exponential mapping and its inverse have a particularly simple form when



(a) Synthetic DT image.

(b) Rician noise with $\sigma = 90$.

(c) Algorithm 1 (72.2 sec).

(d) Algorithm 2 (205.0 sec).

(e) Fast variant of Algorithm 2 (129.8 sec).

Figure 2. *ℓ^2-TV regularization of a diffusion tensor image with high noise level. Algorithm 1 as well as Algorithm 2 and its fast variant converge to the same solution. The TV regularization ($\alpha = 0.70$) removes almost all the noise, and it preserves the sharp transitions. The signal-to-noise-ratio improvement is $\Delta\mathrm{SNR} = 19.03$ in all three cases. The numbers in brackets denote the CPU time for 4000 iterations.*

regarding S^1 as a unit circle in the complex plane. Then the exponential mapping is given by

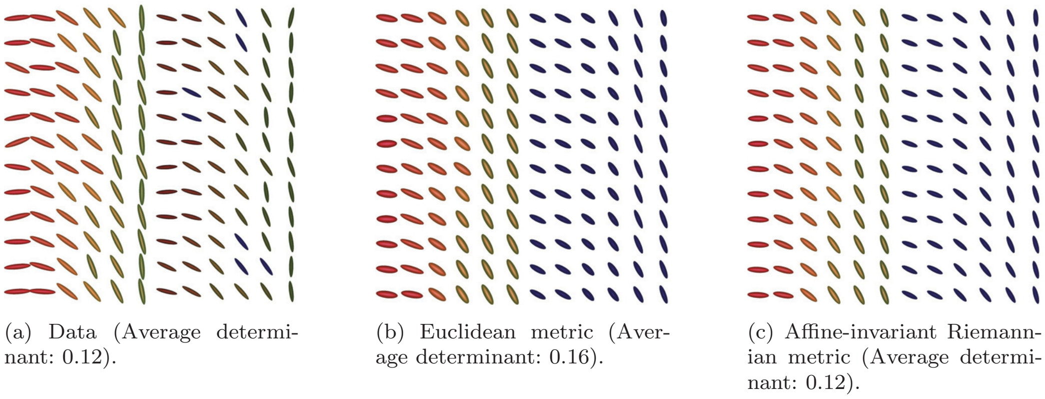
$$\exp_a(v) = e^{i(\theta+v)},$$

where $a = e^{i\theta}$ and $v \in]-\pi;\pi[$. For two nonantipodal points a and b the inverse exponential map reads as

$$\exp_a^{-1}(b) = \arg(b/a),$$

which is the polar angle of the complex number b/a. The distance between two points on the sphere reads as $d(a,b) = |\arg(b/a)|$.

In Figure 4, we apply TV denoising to a real InSAR image taken from [72] originating from ESA. This experiment shows the different effects of TV regularization using different data terms. We use ℓ^2 and ℓ^1 terms as well as the Huber term (with the parameters $\tau = \sqrt{2}$ and $\omega = 1$ in the definition of the Huber function (2.25)). We used 600 iterations of the cyclic proximal point algorithm. We observe that TV regularization reduces the noise significantly. The ℓ^1 data term and the Huber data term appear to be more robust to outliers than the ℓ^2 data term.


(a) Data (Average determinant: 0.12).

(b) Euclidean metric (Average determinant: 0.16).

(c) Affine-invariant Riemannian metric (Average determinant: 0.12).

Figure 3. *TV regularization using the Euclidean norm induces a slight swelling of the diffusion tensors; that is, the average determinant of the tensors in* (b) *is higher than that of the data. The affine-invariant Riemannian metric used in this paper is less affected by the swelling effect.*

4.3. $\mathbb{R}^2 \times S^1$–valued images: Denoising in LCh color space. It was observed that TV-based denoising may give better results when using certain nonflat color models instead of the classical RGB color space [23]. One of these nonflat models is the HSV color space, which leads to cylindrical data living in the product space $\mathbb{R}^2 \times S^1$.

We here use the LCh color space. Similar to the HSV space it is a cylindrical space consisting of a luminance component $L \in \mathbb{R}_0^+$, a chroma component $C \in \mathbb{R}_0^+$, and a hue component $h \in S^1$. The difference between HSV and LCh is that the first derives directly from the RGB space, whereas the latter derives from the Lab color space (also called $L^*a^*b^*$ space), which is intended to better match the human visual perception than the technically motivated RGB space. We perform the color space conversions using the built-in functions of MATLAB. For the hue- and range-preserving enhancement of color images see [66].

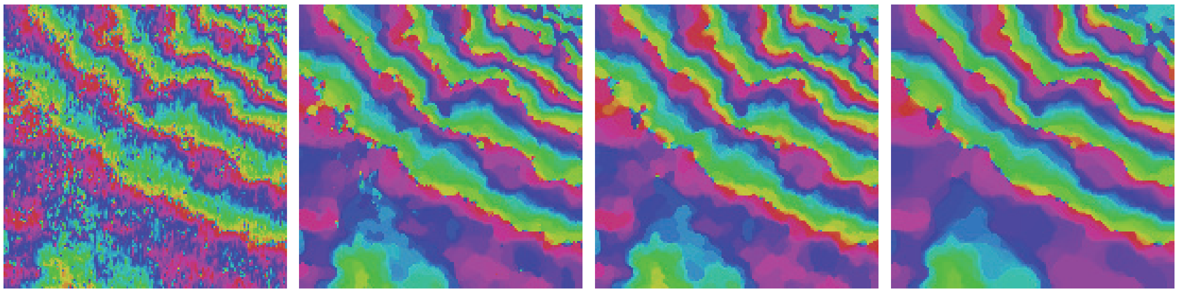
The exponential and the logarithmic mappings are given componentwise by the respective mappings on $\mathbb{R}^2$ and S^1. Note that in spite of this separability property, the proposed algorithm is not equivalent to performing the algorithm on $\mathbb{R}^2$ and S^1 separately (except for $p = q = 2$). The reason is that the path length calculated according to Tables 1 and 2 except for $p, q = 2$ depends nonlinearly on the distance in the product manifold.

In Figure 5 we compare denoising in the RGB space with denoising in the LCh space. The RGB example was computed using the split Bregman method for TV denoising, which is a state-of-the-art method for vectorial TV regularization [46, 42]. We optimized the corresponding model parameter with respect to the peak signal-to-noise ratio (PSNR) given by

$$\mathrm{PSNR}(x) = 10 \log_{10}\left(\frac{3mn \cdot (\max_{i,j,k} |g_{i,j,k}|)^2}{\sum_{i,j,k} |g_{i,j,k} - x_{i,j,k}|^2} \right),$$

where $g \in \mathbb{R}^{n \times m \times 3}$ denotes the ground truth (in RGB space). In Figure 5 we observe that the LCh color space denoising can indeed lead to better results than the vectorial RGB denoising.

4.4. S^2-valued images. We next apply our methods to images taking values in the 2D sphere S^2. For example, spherical data appear in image processing in the context of



(a) InSAR image (real data). (b) ℓ^2-TV ($\alpha = 0.60$). (c) ℓ^1-TV ($\alpha = 0.60$). (d) TV with Huber data term ($\alpha = 0.60$).

Figure 4. *TV denoising of an InSAR image of dimension* 150×150. *The* S^1*-valued data are visualized as hue component in the HSV color space. TV minimization reliably removes the noise while preserving the structure of the image. We observe that* ℓ^1 *and Huber data terms are slightly more robust to outliers. In all three cases the runtime is about* 20 *sec for* 600 *iterations.*

chromaticity-based color models [23, 84] and as orientation fields of 3D images [71].

For a unit vector a on the unit sphere S^2 in $\mathbb{R}^3$ and a nonzero tangent vector v to the sphere at the point a, the exponential mapping is given by

$$\exp_a(v) = a \cdot \cos\|v\| + \frac{a \cdot \sin\|v\|}{\|v\|}.$$

The inverse $\exp_a^{-1}$ of the exponential mapping is well defined for nonantipodal points a and b and is given by

$$\exp_a^{-1}(b) = \arccos(\langle a, b\rangle) \cdot \frac{b - \langle b, a\rangle\, a}{\|b - \langle b, a\rangle\, a\|},$$

where $\langle\cdot,\cdot\rangle$ denotes the standard inner product in $\mathbb{R}^3$. The distance between a and b is $d(a,b) = \arccos(\langle a, b\rangle)$.

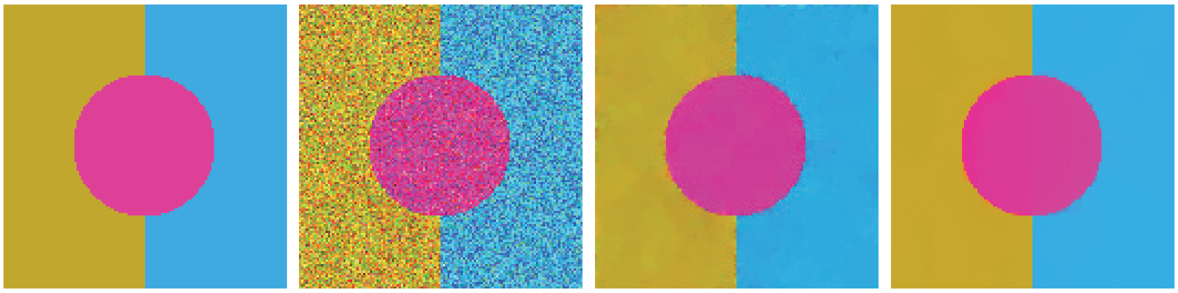
We test the denoising potential of our algorithm on a (synthetic) spherical-valued image. In the context of directional statistics a popular noise model on S^2 uses the von Mises–Fisher distribution having the probability density

$$f(x) = c(\kappa)\exp(\kappa\,\langle x, \mu\rangle).$$

Here the parameter $\kappa > 0$ expresses the concentration around the mean orientation $\mu \in S^2$—the higher the κ, the more concentrated the distribution. The constant $c(\kappa)$ is used for normalization to obtain a probability measure. For each data point $x_{ij} \in S^2$, we consider the above distribution with $\mu = x_{ij}$ and draw a sample. For the simulation of the distribution we used the implementation [53]; see [87] for a description of the algorithm. In Figure 6, we observe that the noise is almost completely removed by TV minimization and that the edges are retained.

4.5. SO(3) data.

Measurements which involve the orientations of rigid objects in 3D space lead to data which take their values in the rotation group SO(3). Examples of SO(3)-valued data are aircraft orientations [83] and protein alignments [48]. They also appear in the



(a) Original image. (b) Gaussian noise (PSNR: 15.64). (c) ℓ^2-TV in RGB space ($\alpha = 0.22$, PSNR: 23.92). (d) ℓ^2-TV in LCh space ($\alpha = 0.90$, PSNR: 32.13).

Figure 5. *TV denoising in different color spaces. We see that measuring the distance in the nonflat LCh metric can lead to higher reconstruction quality than in the RGB color space. The runtime is* 13 *sec for* 1000 *iterations.*

context of tracking 3D rotational data arising in robotics [34]; see also [61] for connections with directional statistics.

The special orthogonal group SO(3) consists of all orthogonal 3×3 matrices with determinant one, i.e.,

$$\mathrm{SO}(3) = \left\{ Q \in GL(3) \ : \ Q^t Q = I_3, \det Q = 1 \right\}.$$

As usual for matrix groups, we consider only the tangent space to SO(3) in the identity matrix I_3. It is given by the space of 3×3 skew-symmetric matrices so(3). Identifying the tangent space at an arbitrary point P with the tangent space at I_3 (via the differential of the left group action) the exponential mapping $\exp_P : \mathrm{so}(3) \to \mathrm{SO}(3)$ in the point P is given by

$$\exp_P(W) = \exp(W)P.$$

Here exp denotes the matrix exponential. For $P, Q \in \mathrm{SO}(3)$, the "inverse" of the above exponential mapping reads as

$$\exp_P^{-1}(Q) = \log(QP^t),$$

where log denotes the principal logarithm (which may be viewed as the componentwise principal logarithm on the eigenvalues). The distance between P and Q equals the Frobenius norm of $\log(QP^t)$.

For the matrix operations needed above there are closed form expressions available; see, e.g., [61]. More precisely, to compute the matrix exponential of a skew-symmetric matrix W we use the Rodrigues formula

$$\exp(W) = I_3 + \frac{\sin(a)}{a} W + \frac{1 - \cos a}{a^2} W^2 \qquad \text{for } a = \sqrt{\mathrm{trace}(W^t W)} > 0.$$

For $a = 0$, we have $\exp(W) = I_3$. Concerning the principal matrix logarithm of a rotation matrix P, we let $\theta = \arccos(\mathrm{trace}(P) - 1)/2$. If $\theta = 0$, then $\log(P) = 0$, the zero matrix. For $|\theta| < \pi$, the principal logarithm of P is given by

$$\log(P) = \frac{\theta \cdot Y}{2 \sin(\theta)}, \qquad \text{where } Y = (P - P^t)/2.$$

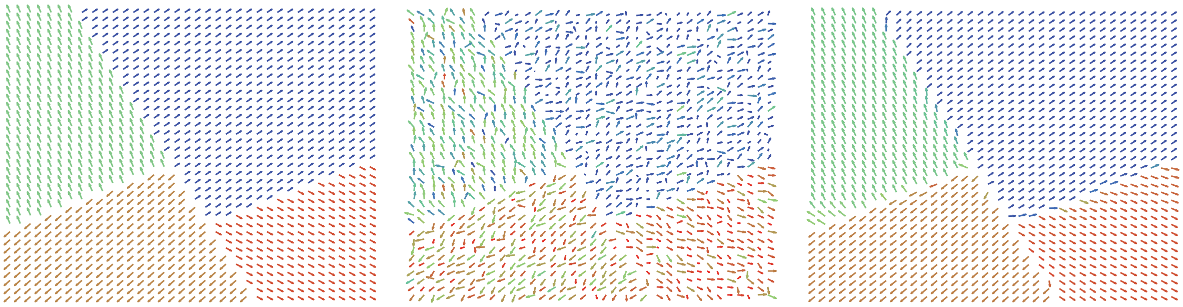


Figure 6. *Denoising of an S^2-valued image. The polar angle is coded both as length of the vectors and as color (red pointing towards the reader, blue away from the reader).* Left: *Original;* Center: *Von Mises–Fisher noise of level $\kappa = 12.7$;* Right: *ℓ^1-TV regularization using $\alpha = 0.5$. The noise is almost completely removed, whereas the jumps are preserved ($\Delta\mathrm{SNR} = 6.85$). The runtime is 12.9 sec for 7000 iterations.*

In Figure 7, we consider a synthetic 1D signal consisting of 130 rotation matrices (visualized as tripods). The signal varies smoothly except for a single jump at the 50th matrix. We simulate noisy data using the matrix Fisher distribution [56] which is given by the density

$$f(X) = c_F \exp\left\{\mathrm{trace}(F^t X)\right\}, \ X \in \mathrm{SO}(3).$$

The matrix F is a fixed 3×3 parameter matrix which describes the mode and the concentration of the distribution. In the isotropic case, the concentration of the distribution can be described by a single parameter $\kappa > 0$ which can be regarded as noise level. A small value of κ corresponds to a high level of noise. Our simulation uses a method recently introduced in [41] and relies on the sampling of quaternions following a related Bingham distribution. To simulate the latter we used the code from [18], which implements the method in [51]. For details we refer the reader to [41, Chap. 5].

The results in Figure 7 show that the proposed algorithm removes the noise. The resulting signal is smoothed while the jump is preserved. In that experiment, we also compare total variation with Huber regularization terms. We see that the Huber regularization exhibits fewer staircasing artifacts than TV regularization.

5. Conclusion and future research.

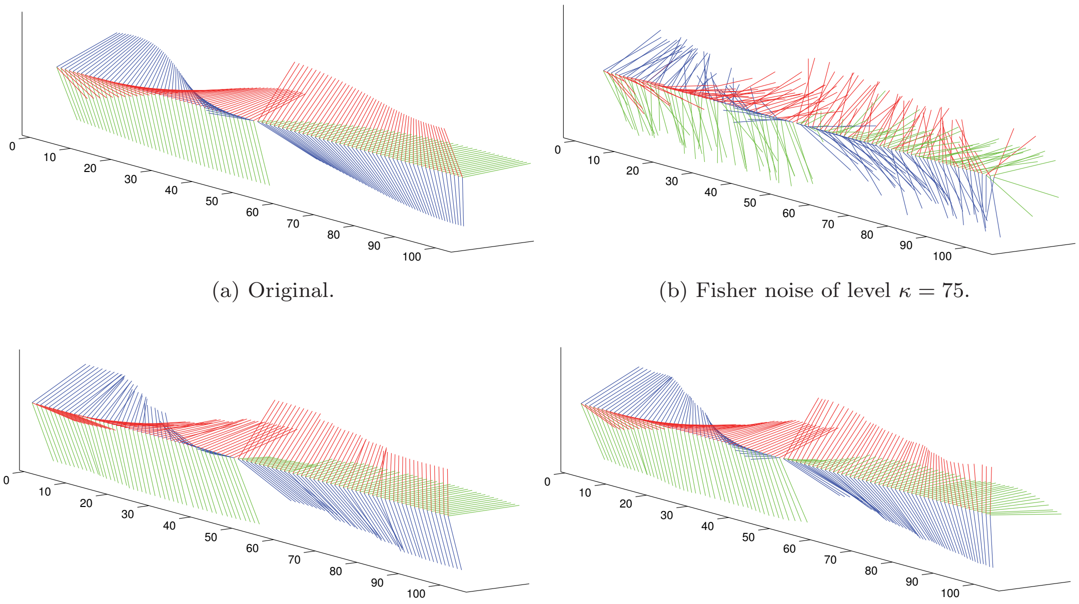
In this paper we have developed proximal point algorithms for TV minimization for manifold-valued data. Our experiments show the denoising capability of the developed algorithms in various manifolds appearing in applications. For Hadamard spaces, we obtain convergence towards a global minimizer of the TV functional.

In future work, we address Blake–Zisserman and Potts functionals for manifold-valued data.

Appendix A. Proof of Proposition 1.

Modifying arguments from section 3, we supply the proof of Proposition 1.

Proof of Proposition 1. We show the statement for the functions G_{ij}. As in the proof of Theorem 2 we see that $y^*_{k,l} = x_{k,l}$ for $k \neq i, i+1$ and $l \neq j$. Then we consider the components y^*_{ij} and $y^*_{i,j+1}$ of a minimizer. We fix a shortest geodesic connecting x_{ij} and $x_{i,j+1}$. We define the points z, z' on this geodesic as in the proof of Theorem 2. Then we may apply the same arguments as in that proof to reduce to the situation where $x_{ij}, z', z, x_{i,j+1}$



(a) Original.

(b) Fisher noise of level $\kappa = 75$.

(c) ℓ^2-TV ($\alpha = 2.0$, $\Delta\text{SNR} = 6.32$).

(d) ℓ^2-Huber ($\alpha = 0.9$, $\Delta\text{SNR} = 9.15$).

Figure 7. *Denoising of a* SO(3)*-valued time-series; TV regularization removes the noise and preserves the jump. The Huber regularization term gives even better results with fewer staircasing effects. The runtimes for TV and Huber regularization are* 1.3 *sec and* 4.7 *sec, respectively, for* 1000 *iterations.*

lie on the geodesic in this ordering. Then the estimate (3.2) is true since we chose a shortest geodesic. This implies $d(z, z') \leq d(y^*_{ij}, y^*_{i,j+1})$. On the other hand, y^* is a minimizer of G_{ij}. This implies $d(y^*_{ij}, y^*_{i,j+1}) \leq d(z, z')$. Hence, equality holds and y^*_{ij} and $y^*_{i,j+1}$ lie on a shortest geodesic connecting x_{ij} and $x_{i,j+1}$. The statements for the functionals H_{ij} and F follow analogously. ■

Acknowledgments. The authors would like to thank Gabriele Steidl for valuable discussions and for pointing out the PPXA algorithm in [29] which led us to consider the parallel proximal algorithms in this paper. They would also like to thank Klaus Hahn for valuable discussions.

REFERENCES

[1] B. Afsari, R. Tron, and R. Vidal, *On the convergence of gradient descent for finding the Riemannian center of mass*, SIAM J. Control Optim., 51 (2013), pp. 2230–2260.

[2] S. Alliney, *Digital filters as absolute norm regularizers*, IEEE Trans. Signal Process., 40 (1992), pp. 1548–1562.

[3] A. Anagaw and M. Sacchi, *Edge-preserving seismic imaging using the total variation method*, J. Geophys. Eng., 9 (2012), pp. 138–146.

[4] M. Arnaudon, C. Dombry, A. Phan, and L. Yang, *Stochastic algorithms for computing means of probability measures*, Stochastic Process. Appl., 122 (2012), pp. 1437–1455.

[5] M. Arnaudon and L. Miclo, *Means in complete manifolds: Uniqueness and approximation*, ESAIM Probab. Stat., 18 (2014), pp. 185–206.

[6] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, *Fast and simple calculus on tensors in the log-*

Euclidean framework, in Medical Image Computing and Computer-Assisted Intervention—MICCAI 2005, Springer, Berlin, 2005, pp. 115–122.

[7] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, *Log-Euclidean metrics for fast and simple calculus on diffusion tensors*, Magn. Reson. Med., 56 (2006), pp. 411–421.

[8] D. Azagra and J. Ferrera, *Proximal calculus on Riemannian manifolds*, Mediterr. J. Math., 2 (2005), pp. 437–450.

[9] M. Bačák, *Computing medians and means in Hadamard spaces*, SIAM J. Optim., 24 (2014), pp. 1542–1566.

[10] W. Ballmann, M. Gromov, and V. Schroeder, *Manifolds of Nonpositive Curvature*, Birkhäuser, Boston, 1985.

[11] P. Basser, J. Mattiello, and D. LeBihan, *MR diffusion tensor spectroscopy and imaging*, Biophys. J., 66 (1994), pp. 259–267.

[12] S. Basu, T. Fletcher, and R. Whitaker, *Rician noise removal in diffusion tensor MRI*, in Medical Image Computing and Computer-Assisted Intervention 2006, Springer, Berlin, 2006, pp. 117–125.

[13] D. Bertsekas, *Incremental proximal methods for large scale convex optimization*, Math. Program., 129 (2011), pp. 163–195.

[14] R. Bhattacharya and V. Patrangenaru, *Large sample theory of intrinsic and extrinsic sample means on manifolds* I, Ann. Statist., 31 (2003), pp. 1–29.

[15] R. Bhattacharya and V. Patrangenaru, *Large sample theory of intrinsic and extrinsic sample means on manifolds* II, Ann. Statist., 33 (2005), pp. 1225–1259.

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn., 3 (2011), pp. 1–122.

[17] M. Bridson and A. Haefliger, *Metric Spaces of Non-positive Curvature*, Springer, Berlin, 1999.

[18] M. Brubaker, M. Salzmann, and R. Urtasun, *A family of MCMC methods on implicitly defined manifolds*, in Proceedings of the International Conference on Artificial Intelligence and Statistics, 2012, pp. 161–172.

[19] A. Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vision, 20 (2004), pp. 89–97.

[20] A. Chambolle and P.-L. Lions, *Image recovery via total variation minimization and related problems*, Numer. Math., 76 (1997), pp. 167–188.

[21] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, J. Math. Imaging Vision, 40 (2011), pp. 120–145.

[22] T. F. Chan and S. Esedoḡlu, *Aspects of total variation regularized L^1 function approximation*, SIAM J. Appl. Math., 65 (2005), pp. 1817–1837.

[23] T. Chan, S. Kang, and J. Shen, *Total variation denoising and enhancement of color images based on the CB and HSV color models*, J. Vis. Comm. Image Represent., 12 (2001), pp. 422–435.

[24] C. Chaux, P. Combettes, J.-C. Pesquet, and V. Wajs, *A variational formulation for frame based inverse problems*, Inverse Problems, 23 (2007), pp. 1495–1518.

[25] C. Chefd'Hotel, D. Tschumperlé, R. Deriche, and O. Faugeras, *Regularizing flows for constrained matrix-valued images*, J. Math. Imaging Vision, 20 (2004), pp. 147–162.

[26] B. Chen and E. Hsu, *Noise removal in magnetic resonance diffusion tensor imaging*, Magn. Res. Med., 54 (2005), pp. 393–401.

[27] T. Chen, W. Yin, X. Zhou, D. Comaniciu, and T. Huang, *Total variation models for variable lighting face recognition*, IEEE Trans. Pattern Anal. Mach. Intell., 28 (2006), pp. 1519–1524.

[28] C. Clason, B. Jin, and K. Kunisch, *A semismooth Newton method for L^1 data fitting with automatic choice of regularization parameters and noise calibration*, SIAM J. Imaging Sci., 3 (2010), pp. 199–231.

[29] P. Combettes and J.-C. Pesquet, *Proximal splitting methods in signal processing*, in Fixed-Point Algorithms for Inverse Problems in Science and Engineering, Springer, New York, 2011, pp. 185–212.

[30] P. Cook, Y. Bai, S. Nedjati-Gilani, K. Seunarine, M. Hall, G. Parker, and D. Alexander, *Camino: Open-source diffusion-MRI reconstruction and processing*, in 14th Scientific Meeting of the International Society for Magnetic Resonance in Medicine, 2006, p. 2759.

[31] N. Dey, L. Blanc-Feraud, C. Zimmer, P. Roux, Z. Kam, J.-C. Olivo-Marin, and J. Zerubia, *Richardson-Lucy algorithm with total variation regularization for 3D confocal microscope deconvolu-*

tion, Micros. Res. Tech., 69 (2006), pp. 260–266.

[32] M. do Carmo, *Riemannian Geometry*, Birkhäuser, Boston, 1992.

[33] Y. Dong, M. Hintermüller, and M. Neri, *An efficient primal-dual method for l^1 TV image restoration*, SIAM J. Imaging Sci., 2 (2009), pp. 1168–1189.

[34] T. Drummond and R. Cipolla, *Real-time visual tracking of complex structures*, IEEE Trans. Pattern Anal. Mach. Intell., 24 (2002), pp. 932–946.

[35] O. Ferreira and P. Oliveira, *Proximal point algorithm on Riemannian manifolds*, Optimization, 51 (2002), pp. 257–270.

[36] R. Ferreira, J. Xavier, J. Costeira, and V. Barroso, *Newton algorithms for Riemannian distance related problems on connected locally symmetric manifolds*, IEEE J. Sel. Top. Sign. Process., 7 (2013), pp. 634–645.

[37] P. Fillard, X. Pennec, V. Arsigny, and N. Ayache, *Clinical DT-MRI estimation, smoothing, and fiber tracking with log-Euclidean metrics*, IEEE Trans. Med. Imag., 26 (2007), pp. 1472–1482.

[38] P. Fletcher, *Geodesic regression and the theory of least squares on Riemannian manifolds*, Int. J. Comput. Vis., 105 (2013), pp. 171–185.

[39] P. Fletcher and S. Joshi, *Riemannian geometry for the statistical analysis of diffusion tensor data*, Signal Process., 87 (2007), pp. 250–262.

[40] P. Fletcher, C. Lu, S. Pizer, and S. Joshi, *Principal geodesic analysis for the study of nonlinear statistics of shape*, IEEE Trans. Med. Imag., 23 (2004), pp. 995–1005.

[41] A. Ganeiber, *Estimation and Simulation in Directional and Statistical Shape Models*, Ph.D. thesis, University of Leeds, Leeds, UK, 2012.

[42] P. Getreuer, *Rudin-Osher-Fatemi total variation denoising using split Bregman*, Image Processing On Line, 2012, http://dx.doi.org/10.5201/ipol.2012.g-tvd.

[43] M. Giaquinta, G. Modica, and J. Souček, *Variational problems for maps of bounded variation with values in S^1*, Calc. Var., 1 (1993), pp. 87–121.

[44] M. Giaquinta and D. Mucci, *The BV-energy of maps into a manifold: Relaxation and density results*, Ann. Sc. Norm. Super. Pisa Cl. Sci. (5), 5 (2006), pp. 483–548.

[45] M. Giaquinta and D. Mucci, *Maps of bounded variation with values into a manifold: Total variation and relaxed energy*, Pure Appl. Math. Q., 3 (2007), pp. 513–538.

[46] T. Goldstein and S. Osher, *The split Bregman method for L^1-regularized problems*, SIAM J. Imaging Sci., 2 (2009), pp. 323–343.

[47] Y. Gousseau and J.-M. Morel, *Are natural images of bounded variation?*, SIAM J. Math. Anal., 33 (2001), pp. 634–648.

[48] P. Green and K. Mardia, *Bayesian alignment using hierarchical models, with applications in protein bioinformatics*, Biometrika, 93 (2006), pp. 235–254.

[49] P. Grohs, H. Hardering, and O. Sander, *Optimal a priori discretization error bounds for geodesic finite elements*, Found. Comput. Math., DOI 10.1007/s10208-014-9230-z, 2014.

[50] P. Grohs and J. Wallner, *Interpolatory wavelets for manifold-valued data*, Appl. Comput. Harmon. Anal., 27 (2009), pp. 325–333.

[51] P. Hoff, *Simulation of the matrix Bingham-von Mises-Fisher distribution, with applications to multivariate and relational data*, J. Comput. Graph. Statist., 18 (2009), pp. 438–456.

[52] H. Johansen-Berg and T. Behrens, *Diffusion MRI: From Quantitative Measurement to In-Vivo Neuroanatomy*, Academic Press, London, 2009.

[53] S. Jung, *Random Number Generation from von Mises–Fisher Distribution*, http://www.stat.pitt.edu/sungkyu/MiscPage.html/ (2010).

[54] H. Karcher, *Riemannian center of mass and mollifier smoothing*, Comm. Pure Appl. Math., 30 (1977), pp. 509–541.

[55] W. Kendall, *Probability, convexity, and harmonic maps with small image* I: *Uniqueness and fine existence*, Proc. London Math. Soc. (3), 61 (1990), pp. 371–406.

[56] C. Khatri and K. Mardia, *The von Mises-Fisher matrix distribution in orientation statistics*, J. Roy. Statist. Soc. Ser. B, 39 (1977), pp. 95–106.

[57] R. Kimmel and N. Sochen, *Orientation diffusion or how to comb a porcupine*, J. Vis. Comm. Image Represent., 13 (2002), pp. 238–248.

[58] R. Lai and S. Osher, *A splitting method for orthogonality constrained problems*, J. Sci. Comput., 58

(2014), pp. 431–449.

[59] D. Le Bihan, J.-F. Mangin, C. Poupon, C. Clark, S. Pappata, N. Molko, and H. Chabriat, *Diffusion tensor imaging: Concepts and applications*, J. Magn. Reson. Imag., 13 (2001), pp. 534–546.

[60] D. Massonnet and K. Feigl, *Radar interferometry and its application to changes in the earth's surface*, Rev. Geophys., 36 (1998), pp. 441–500.

[61] M. Moakher, *Means and averaging in the group of rotations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 1–16.

[62] J.-J. Moreau, *Fonctions convexes duales et points proximaux dans un espace hilbertien*, C. R. Acad. Sci. Paris Sér. A Math., 255 (1962), pp. 2897–2899.

[63] M. Nikolova, *Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers*, SIAM J. Numer. Anal., 40 (2002), pp. 965–994.

[64] M. Nikolova, *A variational approach to remove outliers and impulse noise*, J. Math. Imaging Vision, 20 (2004), pp. 99–120.

[65] M. Nikolova, M. K. Ng, S. Zhang, and W.-K. Ching, *Efficient reconstruction of piecewise constant images using nonsmooth nonconvex minimization*, SIAM J. Imaging Sci., 1 (2008), pp. 2–25.

[66] M. Nikolova and G. Steidl, *Fast hue and range preserving histogram specification: Theory and new algorithms for color image enhancement*, IEEE Trans. Image Process., 23 (2014), pp. 4087–4100.

[67] J. Oller and J. Corcuera, *Intrinsic analysis of statistical estimation*, Ann. Statist., 23 (1995), pp. 1562–1581.

[68] X. Pennec, *Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements*, J. Math. Imaging Vision, 25 (2006), pp. 127–154.

[69] X. Pennec, P. Fillard, and N. Ayache, *A Riemannian framework for tensor computing*, Int. J. Comput. Vis., 66 (2006), pp. 41–66.

[70] P. Petrushev, A. Cohen, H. Xu, and R. DeVore, *Nonlinear approximation and the space* $BV(\mathbb{R}^2)$, Amer. J. Math., 121 (1999), pp. 587–628.

[71] R. Rezakhaniha, A. Agianniotis, J. Schrauwen, A. Griffa, D. Sage, C. Bouten, F. Van de Vosse, M. Unser, and N. Stergiopulos, *Experimental investigation of collagen waviness and orientation in the arterial adventitia using confocal laser scanning microscopy*, Biomech. Model. Mechan., 11 (2012), pp. 461–473.

[72] F. Rocca, C. Prati, and A. Ferretti, *An overview of SAR interferometry*, in Proceedings of the 3rd ERS Symposium on Space at the Service of our Environment, Florence, 1997; also available online from http://earth.esa.int/workshops/ers97/program-details/speeches/rocca-et-al.

[73] G. Rosman, M. Bronstein, A. Bronstein, A. Wolf, and R. Kimmel, *Group-valued regularization framework for motion segmentation of dynamic non-rigid shapes*, in Scale Space and Variational Methods in Computer Vision, Springer, Berlin, 2012, pp. 725–736.

[74] G. Rosman, Y. Wang, X.-C. Tai, R. Kimmel, and A. Bruckstein, *Fast regularization of matrix-valued images*, in Efficient Algorithms for Global Optimization Methods in Computer Vision, Springer, Berlin, 2014, pp. 19–43.

[75] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.

[76] S. Setzer, G. Steidl, and T. Teuber, *On vector and matrix median computation*, J. Comput. Appl. Math., 236 (2012), pp. 2200–2222.

[77] G. Steidl, J. Weickert, T. Brox, P. Mrázek, and M. Welk, *On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and SIDEs*, SIAM J. Numer. Anal., 42 (2004), pp. 686–713.

[78] E. Strekalovskiy and D. Cremers, *Total variation for cyclic structures: Convex relaxation and efficient minimization*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 1905–1911.

[79] D. Strong and T. Chan, *Edge-preserving and scale-dependent properties of total variation regularization*, Inverse Problems, 19 (2003), pp. S165–S187.

[80] K.-T. Sturm, *Probability measures on metric spaces of nonpositive curvature*, in Heat Kernels and Analysis on Manifolds, Graphs, and Metric Spaces, Contemp. Math. 338, American Mathematical Society, Providence, RI, 2003, pp. 357–390.

[81] D. Tschumperlé and R. Deriche, *Diffusion tensor regularization with constraints preservation*, in

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2001, pp. I-948–I-953.

[82] M. Unser and P. Tafti, *An Introduction to Sparse Stochastic Processes*, Cambridge University Press, Cambridge, UK, 2014.

[83] I. Ur Rahman, I. Drori, V. C. Stodden, D. L. Donoho, and P. Schröder, *Multiscale representations for manifold-valued data*, Multiscale Model. Simul., 4 (2005), pp. 1201–1232.

[84] L. A. Vese and S. J. Osher, *Numerical methods for p-harmonic flows and applications to image processing*, SIAM J. Numer. Anal., 40 (2002), pp. 2085–2104.

[85] J. Wallner and N. Dyn, *Convergence and C^1 analysis of subdivision schemes on manifolds by proximity*, Comput. Aided Geom. Design, 22 (2005), pp. 593–622.

[86] A. Weinmann, *Interpolatory multiscale representation for functions between manifolds*, SIAM J. Math. Anal., 44 (2012), pp. 162–191.

[87] A. Wood, *Simulation of the von Mises-Fisher distribution*, Comm. Stat. Simulat. Comput., 23 (1994), pp. 157–164.

[88] J. Yang, Y. Zhang, and W. Yin, *A fast alternating direction method for $TVL1$-$L2$ signal reconstruction from partial Fourier data*, IEEE J. Sel. Top. Sign. Process., 4 (2010), pp. 288–297.

[89] C. Zach, T. Pock, and H. Bischof, *A duality based approach for realtime TV-L^1 optical flow*, in Pattern Recognition, Springer, Berlin, 2007, pp. 214–223.