Fast Regularization of Matrix-Valued Images

Guy Rosman[1*], Yu Wang[1], Xue-Cheng Tai[2],
Ron Kimmel[1], and Alfred M. Bruckstein[1]

[1] Department of Computer Science,
Technion, Haifa 32000, Israel
{rosman,yuwang,ron,freddy}@cs.technion.ac.il
[2] Department of Mathematics
University of Bergen, Johaness Brunsgate 12
5007 Norway.
tai@mi.uib.no

Abstract. Regularization of matrix-valued data is important in many fields, such as medical imaging, motion analysis and scene understanding, where accurate estimation of diffusion tensors or rigid motions is crucial for higher-level computer vision tasks. In this chapter we describe a novel method for efficient regularization of matrix- and group-valued images.
Using the augmented Lagrangian framework we separate the total-variation regularization of matrix-valued images into a regularization and projection steps, both of which are fast and parallelizable. Furthermore we extend our method to a high-order regularization scheme for matrix-valued functions.
We demonstrate the effectiveness of our method for denoising of several group-valued image types, with data in $SO(n)$, $SE(n)$, and $SPD(n)$, and discuss its convergence properties.

Keywords: Regularization, Matrix-manifolds, Lie-groups, Total-variation, Segmentation

1 Introduction

Matrix-valued signals are an important part of computer vision and image processing. Specific fields where matrix-valued data is especially important include tracking and motion analysis [56, 28, 43, 44], robotics [37, 59, 60, 38], image processing and computer vision [42, 40, 63, 10], as well as more general optimization research [63] and 3D reconstruction [10].

Developing efficient regularization schemes for matrix-valued images is an important aspect of analysis and processing in these fields. These images have been an integral part of various domains, such as image processing [4, 52, 25, 58, 62, 11, 15, 39, 48], motion analysis [56, 31, 43], and surface segmentation [44].

We present an efficient method for augmented Lagrangian smoothing of maps from a Cartesian domain into matrix manifolds such as $SO(n)$, $SE(n)$ and $SPD(n)$, the

* This research was supported by European Community's FP7-ERC program, grant agreement no. 267414.

manifolds of special-orthogonal, special-Euclidean, and symmetric, positive-definite, matrices, respectively. Specifically, the data we regularize can be represented as matrices with constraints on their singular values or eigenvalues. The augmented Lagrangian technique allows us in such cases to separate the optimization process into a *total-variation* (TV, [45]) regularization step and an eigenvalue or singular value projection step, both of which are fast and easily parallelizable using consumer graphic processing units (GPUs).

Our method handles each constraint separately via an auxiliary variable. Optimization with respect the separate variables results in a closed-form solution obtained via shrinkage or matrix decomposition, and is efficient to compute. Specifically, the update rule associated with solving the Lie-group auxiliary variable is similar for the case of $SO(n)$, $SE(n)$ and $SPD(n)$, leading to a unified framework which we describe in Sections 3,4. We briefly discuss convergence properties of the suggested algorithms in Section 5. In Section 6 we demonstrate a few results of our method, including motion analysis from depth sensors, direction diffusion, and DTI denoising and reconstruction. Section 7 concludes the paper.

2 A Short Introduction to Lie-Groups

Lie-groups are differentiable manifolds endows with an algebraic group structure, with smooth generators. Lie-groups and their structure have been used extensively in computer vision, and have been the subject of intense research efforts, involving statistics of matrix-valued data [39], and regularization of matrix-valued images [53, 20], as well as describing the evolution of differential processes with Lie-group data [12, 24]. We give a short introduction to Lie-groups in this section and refer the reader to the literature for an in-depth discussion [21, 50].

Because of the group nature of Lie-groups, elements can be mapped via multiplication with their inverse, to the origin. This provides us with a diffeomorphically mapping each points and its neighborhood onto a neighborhood of the origin element, by group action with their inverse, to the identity element of the group. The tangent space in the origin therefore defines a canonical way of parameterizing small changes of the manifold elements via a vector space. Such a vector space is known as the *Lie-algebra* corresponding to the Lie-group. Lie-algebras are equipped with an anti-symmetric bilinear operator, the *Lie-bracket*, that describes the non-commutative part of the group product. Lie-brackets are used in tracking [6], robotics, and computer vision [35], among other applications.

We deal with two Lie-groups, and two related matrix manifolds in this work. The Lie-groups mentioned are

The rotations group $SO(n)$ - The group $SO(n)$ describes all rotations of the n-dimensional Euclidean space. Elements of this group can be described in a matrix form

$$SO(n) = \left\{ \mathbf{R} \in \mathbb{R}_{n\times n}, \mathbf{R}^T\mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1 \right\}, \tag{1}$$

with the group product being matrix multiplication. The Lie-algebra of this group is the space $so(n)$, which can be described by the set of skew-symmetric matrices,

$$so(n) = \left\{ \mathbf{A} \in \mathbb{R}_{n\times n}, \mathbf{A}^T = -\mathbf{A} \right\}. \tag{2}$$

Another manifolds which are of interest and are highly related to $SO(n)$ are its quotient manifolds, the Stiefel manifolds.

The special-Euclidean group $SE(n)$ - This group represents rigid transformations of the n-dimensional Euclidean space. This group can be thought of as the product manifold of $SO(n)$ and the manifold $\mathbb{R}^n$ describing all translations of the Euclidean space. In matrix form this group can be written as

$$SE(n) = \left\{ \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix}, \mathbf{R} \in SO(n), \mathbf{t} \in \mathbb{R}^n \right\}, \tag{3}$$

with matrix multiplication as the group action.

The Lie-algebra of this group can be written as

$$se(n) = \left\{ \begin{pmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0} & 0 \end{pmatrix}, \mathbf{A} \in so(n), \mathbf{t} \in \mathbb{R}^n \right\}, \tag{4}$$

We note that these groups have trivially-defined embeddings into Euclidean spaces, and an easily computable projection operator from the embedding space onto the group. Also, the embedding space we relate to is equipped with a norm: $\|\cdot\|$ denote the Frobenius norm in this chapter. The inner product used in this chapter is also the inner product corresponding to the Frobenius norm – $\langle A, B \rangle = trace\{A^T B\}$. Matrix manifolds for which there exists a simple projector operator include

Symmetric positive definite matrices $SPD(n)$ - This matrix set has been studied extensively in control theory [18], as well as in the context of diffusion tensor images [39], where the matrices are used to describe the diffusion coefficients along each direction. By definition, this group is given in matrix form as

$$SPD(n) = \left\{ \mathbf{A} \in \mathbb{R}_{n \times n}, \mathbf{A} \succeq 0 \right\}, \tag{5}$$

Stiefel manifolds - The Stiefel manifold $\mathbf{V}_k(\mathbb{R}^n)$ is defined as the set of all k-frames in $\mathbb{R}^n$. This can be written as the set of all $n \times k$ matrices with orthonormal columns. This set, too, has a projection operator similar to $SO(n)$.

3 An Augmented Lagrangian Regularization Algorithm for Matrix-valued Images

The optimization problem we consider is the equivalent of the total-variation regularization of a map from the image domain to the matrix-manifold or Lie-group, $\mathcal{G}$ [20],

$$\underset{u\mathcal{G}}{\operatorname{argmin}} \int \|u^{-1}\nabla u\| + \lambda \|u - u_0\|^2 dx. \tag{6}$$

The function u represents an element in an embedding of $\mathcal{G}$ into Euclidean space, specifically for the manifolds $SO(n), SE(n), SPD(n), \mathbf{V}_k(\mathbb{R}^n)$. Elements of $SO(n)$ and $\mathbf{V}_k(\mathbb{R}^n)$ can be embedded into $\mathbb{R}^{n^2}$ and $\mathbb{R}^{nk}$, respectively. Elements of $SE(n)$ can similarly be embedded into $\mathbb{R}^{(n+1)^2}$, or more precisely, an $n(n+1)$-dimensional linear subspace of $\mathbb{R}^{(n+1)^2}$. The elements of $SPD(n)$ can be embedded into $\mathbb{R}^{n(n+1)/2}$. We note that different choice of effectively parametrizing the manifold are possible, simply by making the norm in Equation 6 a weighted one. Specific choices of metric has been discussed in [37, 60], but currently no single canonical choice prevails. Choosing an optimal parameterization is beyond the scope of this work.

We first describe our method in the context of $\mathcal{G} = SO(n)$, and then detail the differences required when $\mathcal{G} = SE(n)$ and $\mathcal{G} = SPD(n)$.

We use the same notation for representation of the manifold point, its matrix representation, and its embedding into the embedding space, as specified in each case we explore.

The term $\|u^{-1}\nabla u\|$ can be thought of as a regularization term placed on elements of the Lie-algebra about each pixel. In order to obtain a fast regularization scheme, we look instead at regularization of an embedding of the Lie-group elements into Euclidean space,

$$\underset{u:\Omega\to\mathcal{G}}{\operatorname{argmin}} \int_\Omega \|\nabla u\| + \lambda \|u-u_0\|^2 dx. \tag{7}$$

The rationale behind the different regularization term $\|\nabla u\|$ stems from the fact that $SO(n)$ and $SE(n)$ are isometries of Euclidean space. In fact, denote by u^j vectors in $\mathbb{R}^n$ representing the columns of the matrix $u(x)$. Since $u(x)$ is approximately an isometry of $\mathbb{R}^n$, let $\Delta\lambda(x)$ denote the maximal local perturbation of the singular values of $u^{-1}(x)$. We assume $\Delta\lambda < 1$. In this case,

$$\begin{aligned} &\left| \left\| \tfrac{\partial}{\partial x_i} u \right\|_F^2 - \left\| u^{-1} \tfrac{\partial}{\partial x_i} u \right\|_F^2 \right| \\ &= \left| \textstyle\sum_{j=1}^n \left\| u^j_{x_i} \right\|^2 - \sum_{j=1}^n \left\| u^{-1} u^j_{x_i} \right\|^2 \right| \\ &\leq \Delta\lambda \textstyle\sum_{j=1}^n \left\| u^j_{x_i} \right\|^2 = \Delta\lambda \left\| u_{x_i} \right\|_F^2 \end{aligned} \tag{8}$$

Hence, as long as the constraint $u(x) \in \mathcal{G}\ \ \forall x \in \Omega$ is approximately fulfilled for an isometry group $\mathcal{G}$, $\|\nabla u\|_F^2 \approx \|u^{-1}\nabla u\|_F^2$. Moreover, such a regularization is possible whenever the data consists of nonsingular and rectangular matrices, and has been used also for SPD matrices [57]. Next, instead of restricting u to $\mathcal{G}$, we add an auxiliary variable, v, at each point, such that $u = v$, and restrict v to $\mathcal{G}$. The equality constraint is enforced via augmented Lagrangian terms [22, 41]. The suggested augmented Lagrangian optimization now reads

$$\begin{aligned} &\min_{v\in\mathcal{G},u\in\mathbb{R}^m} \max_\mu \mathcal{L}(u,v;\mu) \\ &= \min_{v\in\mathcal{G},u\in\mathbb{R}^m} \max_\mu \int \begin{bmatrix} \|\nabla u\| + \lambda\|u-u_0\|^2 + \\ \frac{r}{2}\|v-u\|^2 + \langle \mu, v-u \rangle \end{bmatrix} dx. \end{aligned} \tag{9}$$

Given a fixed Lagrange multiplier μ, the minimization w.r.t. u, v can be split into alternating minimization steps as described in the following two subsections.

3.1 Minimization w.r.t. v

The minimization w.r.t. v is a projection problem per pixel,

$$\begin{aligned} &\operatorname{argmin}_{v\in\mathcal{G}} \tfrac{r}{2}\|v-u\|^2 + \langle \mu, u-v\rangle \\ &= \operatorname{argmin}_{v\in\mathcal{G}} \tfrac{r}{2}\left\|v-\left(\tfrac{\mu}{r}+u\right)\right\|^2 \qquad (10)\\ &= \operatorname{Proj}_{\mathcal{G}}\left(\tfrac{\mu}{r}+u\right), \qquad (11)\end{aligned}$$

where $\operatorname{Proj}_{\mathcal{G}}(\cdot)$ denotes a projection operator onto the specific matrix-group $\mathcal{G}$, and its concrete form for $SO(n)$,$SE(n)$ and $SPD(n)$ will be given later on.

3.2 Minimization w.r.t. u

Minimization with respect to u is a vectorial TV denoising problem

$$\underset{u\in\mathbb{R}^m}{\operatorname{argmin}} \int \|\nabla u\| + \tilde{\lambda}\,\|u-\tilde{u}\,(u_0,v,\mu,r)\|^2\,dx, \qquad (12)$$

with $\tilde{u} = \frac{2\lambda u_0 + rv + 2\mu}{2\lambda + r}$. This problem can be solved via fast minimization techniques – specifically, we chose to use the augmented-Lagrangian TV denoising algorithm [51], as we now describe. In order to obtain fast optimization of the problem with respect to u, we add an auxiliary variable p, along with a constraint that $p = \nabla u$. Again, the constraint is enforced in an augmented Lagrangian manner. The optimal u now becomes a saddle point of the optimization problem

$$\min_{\substack{u\in\mathbb{R}^m\\ p\in\mathbb{R}^{2m}}} \max_{\mu_2} \int \begin{bmatrix} \|p\| + \tilde{\lambda}\,\|u-\tilde{u}\,(u_0,v,\mu,r)\|^2 \\ +\mu_2^T(p-\nabla u) + \frac{r_2}{2}\|p-\nabla u\|^2 \end{bmatrix} dx, \qquad (13)$$

We solve for u using the Euler-Lagrange equation,

$$2\tilde{\lambda}(u-\tilde{u}) + (\operatorname{div}\mu_2 + r_2\operatorname{div}p) + r_2\Delta u = 0, \qquad (14)$$

for example, in the Fourier domain, or by Gauss-Seidel iterations.

The auxiliary field p is updated by rewriting the minimization w.r.t. p as

$$\underset{p\in\mathbb{R}^{2m}}{\operatorname{argmin}} \int \|p\| + \mu_2^T p + \frac{r_2}{2}\|p-\nabla u\|^2, \qquad (15)$$

with the closed-form solution [61, 51]

$$p = \frac{1}{r_2}\max\left(1-\frac{1}{\|w\|},0\right)w, w = r_2\nabla u - \mu_2. \qquad (16)$$

Hence, the main part of the proposed algorithm is to iteratively update v, u, and p respectively. Also, according to the optimality conditions, the Lagrange multipliers μ and μ_2 should be updated by taking

$$\begin{aligned} \mu^k &= \mu^{k-1} + r\left(v^k - u^k\right) \qquad (17)\\ \mu_2^k &= \mu_2^{k-1} + r_2\left(p^k - \nabla u^k\right). \end{aligned}$$

Let

$$\mathcal{F}(u,v,p;\mu,\mu_2) = \int \begin{bmatrix} \lambda \left\|u-u_0\right\|^2 + \frac{r_2}{2}\|p-\nabla u\|^2 + \frac{r}{2}\|u-v\|^2 + \\ +\mu^T(u-v) + \mu_2^T(p-\nabla u) + \|p\| \end{bmatrix} dx. \quad (18)$$

the constrained minimization problem in Equation 7 becomes the following saddle-point problem

$$\min_{\substack{v \in \mathcal{G} \\ u \in \mathbb{R}^m \\ p \in \mathbb{R}^{2m}}} \max_{\mu,\mu_2} \mathcal{F}(u,v,p;\mu,\mu_2) \quad (19)$$

An algorithmic description is summarized as Algorithm 1, whose convergence properties are discussed in Section 5.

Algorithm 1 Fast TV regularization of matrix-valued data

1: **for** $k = 1, 2, \ldots,$ until convergence **do**
2: Update $u^k(x), p^k(x)$, according to Equations (14,16).
3: Update $v^k(x)$, by projection onto the matrix group,

- For $SO(n)$ matrices, according to Equation (20).
- For $SE(n)$ matrices, according to Equation (21).
- For $SPD(n)$ matrices, according to Equation (22).

4: Update $\mu^k(x), \mu_2^k(x)$, according to Equation (17).
5: **end for**

3.3 Regularization of maps onto $SO(n)$

In the case of $\mathcal{G} = SO(n)$, although the embedding of $SO(n)$ in Euclidean space is not a convex set, the projection onto the matrix manifold is easily achieved by means of the singular value decomposition [19]. Let $\mathbf{USV}^T = \left(\frac{\mu}{r} + u^k\right)$ be the SVD decomposition of $\frac{\mu}{r} + u^k$, we update v by

$$\begin{gathered} v^{k+1} = \mathrm{Proj}_{SO(n)}\left(\frac{\mu}{r} + u^k\right) = \mathbf{U}(x)\mathbf{V}^T(x), \\ \mathbf{USV}^T = \left(\frac{\mu}{r} + u^k\right). \end{gathered} \quad (20)$$

Other possibilities include using the Euler-Rodrigues formula, quaternions, or the polar decomposition [29]. We note that the non-convex domain $SO(n)$ prevents a global convergence proof of the type shown in Subsection 5.2 for $SPD(n)$. Convergence properties of the algorithm, in the case of $\mathcal{G} = SO(n)$ and $\mathcal{G} = SE(n)$, are discussed in Subsection 5.1.

We also note that the projection via SVD can be used to project matrices onto the Stiefel manifolds [33], themselves quotient groups of $SO(n)$ [55]. Thus, the same algorithm can be used for Stiefel manifolds as well.

3.4 Regularization of maps onto $SE(n)$

In order to regularize images with values in $SE(n)$, we use an embedding into $\mathbb{R}^{n(n+1)}$ as our main optimization variable, u, per pixel.

The projection step w.r.t. v applies only for the n^2 elements of v describing the rotation matrix, leaving the translation component of $SE(n)$ unconstrained.

Specifically, let $v = (v_R, v_t), v_R \in \mathbb{R}^{n^2}, v_t \in \mathbb{R}^n$ denotes the rotation and translation parts of the current solution. Updating v in step 3 of algorithm 1 assumes the form

$$\begin{aligned} v_R^{k+1} = \mathbf{U}(x)\mathbf{V}^T(x), \quad \mathbf{USV}^T = \left(\tfrac{\mu_R}{r} + u_R^k\right) \\ v_t^{k+1} = \left(\tfrac{\mu_t}{r} + u_t^k\right) \\ v^{k+1} = \mathrm{Proj}_{SE(n)}(v^k) = (v_R^{k+1}, v_t^{k+1}). \end{aligned} \tag{21}$$

3.5 Regularization of maps onto $SPD(n)$

The technique described above can be used also for regularizing symmetric positive-definite matrices. A most prominent example for such matrices is that of diffusion tensor images [49, 4, 53, 13, 5, 27, 30]. This includes several attempts to define efficient and physically meaningful regularization techniques for DTI regularization [53, 65, 7]. Many papers dealing with the analysis of DTI rely on the eigenvalue decomposition of the tensor as well, i.e. for tractography [14], anisotropy measurements [64], and so forth. It is not surprising that the intuitive choice of projecting the eigenvalues of the matrices onto the positive half-space is shown to be optimal [23, 9].

When using an augmented Lagrangian approach, the minimization problem w.r.t. v in step 3 of algorithm 1 is therefore solved by projection of eigenvalues,

$$\begin{aligned} v^{k+1} = \mathrm{Proj}_{SPD(n)}(v^k) = \mathbf{U}(x)\,\mathrm{diag}\left(\tilde{\boldsymbol{\lambda}}\right)\mathbf{U}^T(x), \\ \mathbf{U}\,\mathrm{diag}\left(\boldsymbol{\lambda}\right)\mathbf{U}^T = \left(\tfrac{\mu}{r} + u^k\right), \left(\tilde{\boldsymbol{\lambda}}\right)_i = \max\left(\left(\boldsymbol{\lambda}\right)_i, 0\right), \end{aligned} \tag{22}$$

where the matrix U is a unitary one, representing the eigenvectors of the matrix, and the eigenvalues $\left(\tilde{\boldsymbol{\lambda}}\right)_i$ are the positive projection of the eigenvalues $\left(\boldsymbol{\lambda}\right)_i$. Optimization w.r.t. u is done as in the previous cases, as described in Algorithm 1.

Furthermore, the optimization w.r.t. u, v is now over the domain $\mathbb{R}^m \times SPD(n)$, and the cost function is convex, resulting in a convex optimization problem. The convex domain of optimization allows us to formulate a convergence proof for the algorithm similar to the proof by Tseng [54]. This is discussed in Subsection 5.2. An example of using the proposed method for DTI denoising and reconstruction is shown in Section 6.

3.6 A Higher-Order Prior for Group-Valued Images

We note that the scheme we describe is susceptible to the staircasing effect, since it minimizes the total variation of the map u. While one possibility to avoid such artifacts is to incorporate a linear diffusion term into the functional, there exists a much more elegant solution by incorporating a higher-order differential operator into the regularization term. One such possibile higher-order term generalizes the scheme presented by Wu and Tai [66], by replacing the per-element gradient operator with a Hessian operator. The resulting equivalent of Equation 7 becomes

$$\underset{u \in \mathcal{G}}{\operatorname{argmin}} \int \|Hu\| + \lambda \|u - u_0\|^2 dx, \tag{23}$$

where Hu is the per-channel Hessian operator, defined (on two-dimensional domains) by

$$\left(Hu^{(k)}\right)_{i,j} = \begin{pmatrix} \left(D_{xx}^{-+} u^{(k)}\right)_{i,j} & \left(D_{xy}^{++} u^{(k)}\right)_{i,j} \\ \left(D_{yx}^{++} u^{(k)}\right)_{i,j} & \left(D_{yy}^{-+} u^{(k)}\right)_{i,j} \end{pmatrix} \tag{24}$$

The numerical scheme solves the saddle-point problem

$$\min_{\substack{u \in \mathbb{R}^m \\ p \in \mathbb{R}^{4m}, \\ v \in \mathcal{G}}} \max_{\mu_2} \int \begin{bmatrix} \|p\| + \tilde{\lambda} \left\| u - \tilde{u}\left(u_0, v, \mu, r\right) \right\|^2 \\ + \mu_2^T (p - Hu) + \frac{r_2}{2} \|p - Hu\|^2 \end{bmatrix} dx, \tag{25}$$

The update step w.r.t. u as in Equation 14 is easy to modify, resulting in the Euler-Lagrange equation

$$2\tilde{\lambda}(u - \tilde{u}) - (H^* \mu_2 + r_2 H^* p) + r_2 H^* H u = 0, \tag{26}$$

where H^* is the adjoint operator of the Hessian,

$$H^* p^{(k)} = D_{xx}^{+-} \left(p^{(k)}\right)^{11} + D_{xy}^{--} \left(p^{(k)}\right)^{12} + D_{yx}^{--} \left(p^{(k)}\right)^{21} + D_{yy}^{+-} \left(p^{(k)}\right)^{22} \tag{27}$$

The update step w.r.t. p remains similar to Equation 16, and is given by

$$p = \frac{1}{r_2} \max \left(1 - \frac{1}{\|w\|}, 0\right) w, w = r_2 H u - \mu_2. \tag{28}$$

Updates of the variable v and the Lagrange multipliers μ, μ_2 remain the same as in Algorithm 1. As will be shown in Section 6, this regularization term prevents formation of staircasing effects where these are inappropriate.

4 Regularized DTI Reconstruction

There are several possibilities of using the proposed regularization scheme for DTI reconstruction from diffusion-weighted measurements. Instead of adding a fidelity term as in Equation (7), we add a term for fitting the Stejskal-Tanner equations [49], based on a set of measurements describing the diffusion in specific directions, and reconstruct the full diffusion tensor at each voxel. The fitting term can be written as

$$\sum_i \left\| b_i \mathbf{g}_i^T u \mathbf{g}_i - \log\left(\frac{S_i}{S_0}\right)\right\|^2,$$

where b_i and $\mathbf{g}_i$ are the b-values and gradient vectors, u is the diffusion tensor reconstructed at each voxel, and $\frac{S_i}{S_0}$ define the relative signal ratio for each direction at each voxel. The complete minimization problem reads

$$\underset{\substack{v \in SPD(n)\\ u}}{\operatorname{argmin}} \int \sum_i \left\| b_i \mathbf{g}_i^T u \mathbf{g}_i - \log\left(\frac{S_i}{S_0}\right)\right\|^2 + \lambda \|\nabla u\| + \frac{r}{2}\|v-u\|^2 + \langle \mu, v-u\rangle d \tag{29}$$

While the memory requirements seem less favorable for fast optimization, looking closely at the quadratic penalty data term, we see it can be expressed by looking at a fitting term for the Stejskal-Tanner equations ,

$$\sum_i \left\| b_i \mathbf{g}_i^T u \mathbf{g}_i - \log\left(\frac{S_i}{S_0}\right)\right\|^2 = u^T \mathbf{A} u + \mathbf{b}^T u + c, \tag{30}$$

where $\mathbf{A}$ is a constant matrix over the whole volume,

$$\mathbf{A} = \sum_i b_i^2 \begin{pmatrix} g_1^4 & 2g_1^3 g_2 & 2g_1^3 g_3 & g_1^2 g_2^2 & 2g_1^2 g_2 g_3 & g_1^2 g_3^2 \\ 2g_1^3 g_2 & 4g_1^2 g_2^2 & 4g_1^2 g_2 g_3 & 2g_1 g_2^3 & 4g_1 g_2^2 g_3 & 2g_1 g_2 g_3^2 \\ 2g_1^3 g_3 & 4g_1^2 g_2 g_3 & 4g_1^2 g_3^2 & 2g_1 g_2^2 g_3 & 4g_1 g_2 g_3^2 & 2g_1 g_3^3 \\ g_1^2 g_2^2 & 2g_1 g_2^3 & 2g_1 g_2^2 g_3 & g_2^4 & 2g_2^3 g_3 & g_2^2 g_3^2 \\ 2g_1^2 g_2 g_3 & 4g_1 g_2^2 g_3 & 4g_1 g_2 g_3^2 & 2g_2^3 g_3 & 4g_2^2 g_3^2 & 2g_2 g_3^3 \\ g_1^2 g_3^2 & 2g_1 g_2 g_3^2 & 2g_1 g_3^3 & g_2^2 g_3^2 & 2g_2 g_3^3 & g_3^4 \end{pmatrix} \tag{31}$$

and $\mathbf{b}$ is the vector

$$\mathbf{b} = \sum_i b_i \log\left(\frac{S_i}{S_0}\right) \left(2g_1^2\ 4g_1g_2\ 4g_1g_3\ 2g_2^2\ 4g_2g_3\ 2g_3^2\right)^T, \tag{32}$$

and c is the scalar image

$$c = \sum_i \left(\log\left(\frac{S_i}{S_0}\right)\right)^2. \tag{33}$$

We note that, unlike the denoising case, in the reconstruction case it is the data term that couples together the elements of the tensor together. Care must be taken so as to handle this coupled data term.

Reconstruction with the new data term can be computed using several techniques.

– Freezing all elements of the tensor but one, we obtain from the Euler-Lagrange equations pertaining to Equation 29 an update rule for the image, to be computed

in the Fourier domain, or via Gauss-Seidel iterations. While the coupling between the tensor elements (expressed via the non-diagonal matrix $\mathbf{A}$) prevents us from treating each tensor element separately, the optimization w.r.t. each of the elements converges quite rapidly.
– Another possibility is to take a block Gauss-Seidel approach, and optimize each tensor separately, going over all the voxels one-by-one.
– Yet another possibility is to further decouple the TV and data term, using separate variables and constraining them using an augmented Lagrangian approach.

Of the above techniques, we have tried the first one. The reconstruction obtained is the spatially-regularized version of the *linear-least-squares* (LLS) method. One can incorporate a weighted least-squares (WLS, [47]), or nonlinear-least-squares (NLS) [27] data term instead. Combining such data terms and exploring the interaction between the regularization and nonlinear terms is beyond the scope of this work.

5 Convergence Properties of the Algorithm

We now turn to discuss the local convergence of Algorithm 1.

5.1 Local Convergence for $SO(n)$,$SE(n)$ Regularization

Looking at regularization of maps onto $SO(n)$,$SE(n)$, the non-convex nature of the optimization domain in equation 9 makes it difficult to prove global convergence. Furthermore, the nature of the projection operator into $SO(n)$ and $SE(n)$, makes it difficult to ascertain that at some point the sequence of iterants will converge. While showing there exists a converging subsequence of iterants is easy due to the boundedness of the sub-levelsets [54], the discontinuous nature of the projection unto non-convex spaces may cause the algorithm to oscillate, although this behaviour does not appear in practice. In order to avoid such a possibility and allow for an easy proof of convergence, we take a proximal step approach, and slightly modify our algorithm, as suggested by Attouch et al. [3], changing the first two steps of the algorithm into the minimization problems

$$\begin{aligned} u^k &= \operatorname*{argmin}_u \mathcal{F}(u, v^{k-1}, \mu) + \frac{1}{\theta_k}\|u - u^{k-1}\|^2 \\ v^k &= \operatorname*{argmin}_{v\in\mathcal{G}} \mathcal{F}(u^k, v, \mu) + \frac{1}{\theta_k}\|v - v^{k-1}\|^2. \end{aligned} \tag{34}$$

The proof of convergence becomes similar to the one shown by Attouch et al. [3]. Since $\mathcal{F}(u,v) > -\infty$ and $\{\mathcal{F}(u^k, v^k)\}$ is non-increasing, we have that $\mathcal{F}(u^k, v^k)$ converges to some finite value. Furthermore, by induction we can show that the residual converges to 0, providing us with a guarantee of the asympthotic behavior of the process.

The optimization steps in the modified algorithm remain a projection step and total-variation denoising, but with a change in their parameters. For example, the optimal update rule for v becomes

$$\mathrm{argmin}_{v\in SO(n)} \tfrac{r}{2}\|v-u\|^2 + \langle \mu, v-u\rangle + \tfrac{1}{2\theta_k}\|v-v_{k-1}\|^2 + =$$
$$\mathrm{argmin}_{v\in SO(n)} \left(\tfrac{r}{2}+\tfrac{1}{2\theta_k}\right)\|v\|^2 - \langle v, ru+\mu+\tfrac{v_{k-1}}{\theta_k}\rangle + \tfrac{r}{2}\|u\|^2 - \langle \mu, u\rangle + \tfrac{1}{2\theta_k}\|v_{k-1}\|^2 =$$
$$\mathrm{argmin}_{v\in SO(n)} \left(\tfrac{r}{2}+\tfrac{1}{2\theta_k}\right)\left\|v - \tfrac{ru+\mu+\frac{v_{k-1}}{\theta_k}}{r+\frac{1}{\theta_k}}\right\|^2,$$

where $\frac{1}{2\theta_k}$ denotes the coupling between each iterant and its previous value. We stress, however, that in practice the algorithm converges without the above modification quite well.

5.2 Global Convergence for $SPD(n)$ Regularization

For $SPD(n)$ regularization we basically do a coordinate descent on a convex domain [54] and therefore can show global convergence of our method. At each step of the inner iteration, we do a full minimization with respect to the selected variables block u, v and p. Using the notation provided by [54], we can rewrite our functional as

$$\mathcal{F}_{\mu,\mu_2}(u,v,p) = f_0(u,v,p) + f_1(u) + f_2(v) + f_3(p), \tag{35}$$

where

1. f_0 is a convex, smooth, function.
$$f_0(u,v,p) = \frac{r}{2}\|v-u\|^2 + \langle \mu, v-u\rangle + \frac{r_2}{2}\|p-\nabla u\|^2 + \langle \mu_2, p-\nabla u\rangle$$
2. f_1, f_2 and f_3 are convex, lower-semicontinuous, continuous in their effective domain,
$$f_1(u) = \|u-u_0\|^2 \tag{36}$$
$$f_2(v) = 0 \tag{37}$$
$$f_3(p) = \|p\|. \tag{38}$$

By [54, Proposition 1], it can be shown that the alternating minimization will converge to a minimizer of $\mathcal{F}_{\mu,\mu_2}(u,v,p)$. Along the same proof in [67], it can be proved the whole algorithm converges. For completeness we repeat the proof here. The following characterization for the minimizers of functional $\mathcal{F}(u,v,p;\mu,\mu_2)$ will be used. Assume that (u^*,v^*,p^*) is one of the minimizers, and for arbitrary (u',v',p') we have,

$$\begin{aligned} &\lambda\|u^*-u_0\|^2 - \lambda\|u'-u_0\|^2 + r_2(p^*-\nabla u^*, -(\nabla u^*-\nabla u')) \\ &+r(u^*-v^*, u^*-u') + (\mu^*, u^*-u') + (\mu_2^*, -(\nabla u^*-\nabla u')) \le 0 \end{aligned} \tag{39}$$
$$-r(u^*-v^*, v^*-v') - (\mu^*, v^*-v') \le 0 \tag{40}$$
$$\|p^*\| - \|p'\| + r_2(p^*-\nabla u^*, p^*-p') + (\mu_2^*, p^*-p') \le 0 \tag{41}$$

(see [17],p.38 Proposition 2.2)

Theorem 51 *The sequence $(u^k, v^k, p^k; \mu^k, \mu_2^k)$ generated by Algorithm 1 converges to the saddle-point $(u^*, v^*, p^*; \mu^*, \mu_2^*)$ of the functional $\mathcal{F}(u, v, p; \mu, \mu_2)$*

Proof. Let $\bar{u}^k = u^* - u^k$,$\bar{v}^k = v^* - v^k$,$\bar{p}^k = p^* - p^k$, $\bar{\mu}^k = \mu^* - \mu^k$, and $\bar{\mu}_2^k = \mu_2^* - \mu_2^k$ Since $(u^*, v^*, p^*; \mu^*, \mu_2^*)$ is the saddle point of $\mathcal{F}(u, v, p; \mu, \mu_2)$, we have

$$\mathcal{F}(u^*, v^*, p^*; \mu^*, \mu_2^*) \leq \mathcal{F}(u', v', p'; \mu^*, \mu_2^*), \forall u, v, p \tag{42}$$

In particular when $u' = u^k$ (39) still holds

$$\begin{aligned} &\lambda\|u^* - u_0\|^2 - \lambda\|u^k - u_0\|^2 + r_2(p^* - \nabla u^*, -\nabla(u^* - u^k)) \\ &+r(u^* - v^*, u^* - u^k) + (\mu^*, u^* - u^k) + (\mu_2^*, -\nabla(u^* - u^k)) \leq 0 \end{aligned} \tag{43}$$

On the other hand, since the $(u^k, v^k, p^k; \mu^k, \mu_2^k)$ is the minimizer of $\mathcal{F}(u, v, p; \mu^k, \mu_2^k)$, u^k will also satisfy (39) and after substituting $u' = u^*$ we obtain

$$\begin{aligned} &\lambda\|u^k - u_0\|^2 - \lambda\|u^* - u_0\|^2 + r_2(p^k - \nabla u^k, -\nabla(u^k - u^*)) \\ &+r(u^k - v^k, u^k - u^*) + (\mu^k, u^k - u^*) + (\mu_2^k, -\nabla(u^k - u^*)) \leq 0. \end{aligned} \tag{44}$$

Adding the two inequalities yields

$$r_2(\bar{p}^k - \nabla\bar{u}^k, -\nabla\bar{u}^k) + r(\bar{u}^k - \bar{v}^k, \bar{u}^k) + (\bar{\mu}^k, \bar{u}^k) + (\bar{\mu_2}^k, -\nabla\bar{u}^k) \leq 0 \tag{45}$$

Similarly, w.r.t v^*, v^k using the same argument to (40) we have

$$-r(u^* - v^*, v^* - v^k) - (\mu^*, v^* - v^k) \leq 0 \tag{46}$$
$$-r(u^k - v^k, v^k - v^*) - (\mu^k, v^k - v^*) \leq 0 \tag{47}$$

adding two inequalities yields

$$-r(\bar{u}^k - \bar{v}^k, \bar{v}^k) - (\bar{\mu}^k, \bar{v}^k) \leq 0 \tag{48}$$

w.r.t p^*, p^k, the same argument is applied to (41)

$$\|p^*\| - \|p^k\| + r_2(p^* - \nabla u^*, p^* - p^k) + (\mu_2^*, p^* - p^k) \leq 0 \tag{49}$$
$$\|p^k\| - \|p^*\| + r_2(p^k - \nabla u^k, p^k - p^*) + (\mu_2^k, p^k - p^*) \leq 0 \tag{50}$$

thus

$$r_2(\bar{p}^k - \nabla\bar{u}^k, \bar{p}^k) + (\bar{\mu_2}^k, \bar{p}^k) \leq 0 \tag{51}$$

Adding (45), (48) and (51) we have

$$r_2\|\bar{p}^k - \nabla\bar{u}^k\|^2 + r\|\bar{u}^k - \bar{v}^k\|^2 + (\bar{\mu_2}^k, \bar{p}^k - \nabla\bar{u}^k) + (\bar{\mu}^k, \bar{u}^k - \bar{v}^k) \leq 0 \tag{52}$$

By the way of updating multipliers, also note that $u^* = v^*$ and $p^* = \nabla u^*$ we obtain

$$\bar{\mu}^{k+1} = \bar{\mu}^k + r(\bar{u}^k - \bar{v}^k) \tag{53}$$
$$\bar{\mu}_2^{k+1} = \bar{\mu}_2^k + r_2(\bar{p}^k - \nabla\bar{u}^k) \tag{54}$$

therefore by (52) we have

$$\begin{aligned}
&r_2\|\bar{\mu}^{k+1}\|^2 + r\|\bar{\mu}_2^{k+1}\|^2 - r_2\|\bar{\mu}^k\|^2 - r\|\bar{\mu}_2^k\|^2 \\
&= 2rr_2(\bar{\mu}^k, \bar{u}^k - \bar{v}^k) + 2rr_2(\bar{\mu}_2^k, \bar{p}^k - \nabla\bar{u}^k) + r^2r_2\|\bar{u}^k - \bar{v}^k\|^2 + rr_2^2\|\bar{p}^k - \nabla\bar{u}^k\| \\
&\leq -r^2r_2\|\bar{u}^k - \bar{v}^k\|^2 - rr_2^2\|\bar{p}^k - \nabla\bar{u}^k\| \leq 0 \qquad (55)
\end{aligned}$$

This actually implies μ^k and μ_2^k are bounded, and

$$\lim_{k\to\infty} \|p^k - \nabla u^k\| = 0 \qquad (56)$$

$$\lim_{k\to\infty} \|u^k - v^k\| = 0 \qquad (57)$$

With this in mind, it is not hard to show that $(u^k, v^k, p^k; \mu^*, \mu_2^*)$ converge to the saddle-point of the functional

6 Numerical Results

As discussed above, the proposed algorithmic framework is quite general and is suitable for various applications. In this section, several examples from different applications are used to substantiate the effectiveness and efficiency of our algorithm.

6.1 Directions regularization

Analysis of principal directions in an image or video is an important aspect of modern computer vision, in fields such as video surveillance [36, 26, and references therein], vehicle control [16], crowd behaviour analysis [34], and other applications[40].

The input in this problem is a set of normalized / unnormalized direction vectors located throughout the image domain, either in a dense or sparse set of locations. The goal is to obtained a smoothed version of the underlying direction field. Since $SO(2)$ is isomorphic to S^1, the proposed regularization scheme can be used for regularizing directions as well, as we demonstrate. A reasonable choice for a data term would try to align the rotated first coordinate axis with the motion directions in the area,

$$E_{PMD}(U) = \textstyle\sum_{(x_j,y_j)\in\mathcal{N}(i)} \left(U_{1,1}\left(v_j\right)_x + U_{1,2}\left(v_j\right)_y\right), \qquad (58)$$

where $\left(x_j, y_j, \left(v_j\right)_x, \left(v_j\right)_y\right)$ represent a sampled motion particle [34] in the video sequence, and $U_{i,j}$ represent elements of the solution u.

In Figure 1 we demonstrate two sparsely sampled, noisy, motion fields, and a dense reconstruction of the main direction of motion at each point. The data for the direction estimation was corrupted by adding component-wise Gaussian noise. In the first image, the motion field is comprised of 4 regions with a different motion direction at each region. The second image contains a sparse sampling of an expansion motion field of the form

$$\boldsymbol{v}(x,y) = \frac{(x,y)^T}{\|\,(x,y)\,\|}. \qquad (59)$$

Such an expansion field is often observed by forward-moving vehicles. Note that despite the fact that a vanishing point of the flow is clearly not smooth in terms of the motion directions, the estimation of the motion field is still correct.

An example of the higher order regularization term is shown in Figure 2, using the approach suggested in Subsection 3.6. Note the smooth boundaries create due to the sparsely sampled data term – while the TV solution forces staircasing in the solution, the higher order regularization does not.

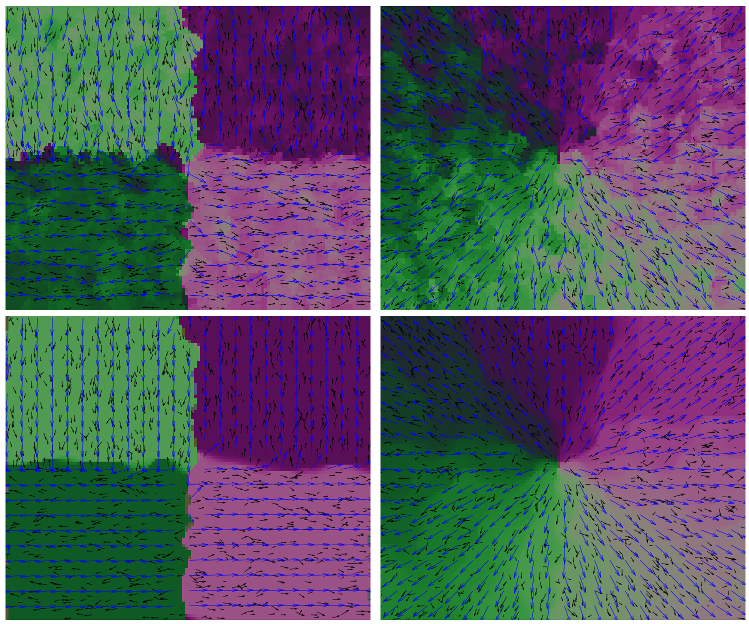

Fig. 1. TV regularization of $SO(n)$ data. Left-to-right, top-to-bottom: the initial estimated field for a 4-piece piecewise constant motion field, a concentric motion field, the denoised images for the piecewise constant field and the concentric motion field. Different colors mark different orientations of the initial/estimated dense field, black arrows signify the measured motion vectors, and blue arrows demonstrate the estimated field after sampling.

In Figure 3 we used the algorithm to obtain a smooth field of principal motion directions over a traffic sequence taken from the UCF crowd flow database [2]. Direction cues are obtained by initializing correlation-based trackers from arbitrary times and positions in the sequence, and observing all of the tracks simultenaously. The result captures the main traffic lanes and shows the viability of our regularization for real data sequences.

Yet another application for direction diffusion is in denoising of directions in fingerprint images. An example for direction diffusion on a fingerprint image taken from the Fingerprint Verification Competition datasets [1] can be seen in Figure 4. Adding a noise of $\sigma = 0.05$ to the image and estimating directions based on the structure ten-

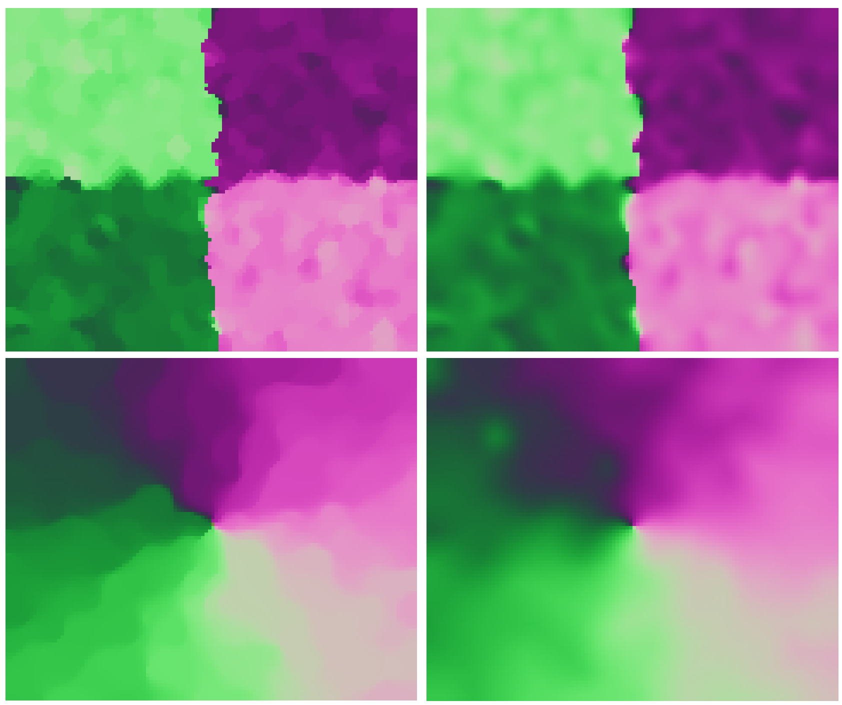

Fig. 2. TV regularization of $SO(n)$ data, based on the same data from Figure 1, with a higher-order regularity term. Different color mark different orientations of the estimated motion field. Left: TV regularization result as demonstrated in Figure 1. Right: regularization results based on Equation 23. The parameter λ was chosen to be 2 for the upper example, and 0.2 for the lower example.

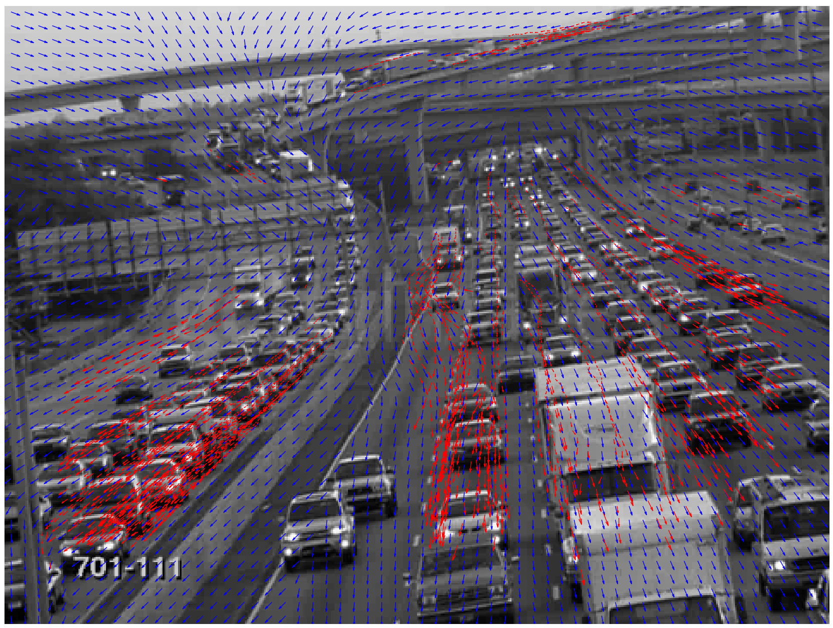

Fig. 3. Regularization of principal motion directions. The red arrows demonstrate measurements of motion cues based on a normalized cross-correlation tracker. Blue arrows demonstrate the regularized directions fields.

sor, we smoothed the direction field and compared it to the field obtained from the original image. We used our method with $\lambda = 3$, and the modified method based on Equation 26 with $\epsilon = 10$, as well as the method suggested by Sochen et al. [46] with $\beta = 100, T = 425$. The resulting MSE values of the tensor field are 0.0317, 0.0270 and 0.0324, respectively, compared to an initial noisy field with $MSE = 0.0449$. These results demonstrate the effectiveness of our method for direction diffusion, even in cases where the staircasing effect may cause unwanted artifacts.

6.2 $SE(n)$ regularization

We now demonstrate a smoothing of $SE(3)$ data obtained from locally matching between two range scans obtained from a Kinect device. For each small surface patch from the depth image we use an *iterative closest point* (ICP) algorithm[8] to match the surface from the previous frame. For each point in the foreground, an ICP algorithm is used to match the point's neighborhood from frame i to that of frame $i - 1$. The background is segmented by simple thresholding. The results from this tracking process over raw range footage are an inherently noisy measurements set in $SE(3)$. We use our algorithm to smooth this $SE(3)$ image, as shown in Figure 5. It can be seen that for a careful choice of the regularization parameter, total variation in the group elements is seen to significantly reduce rigid motion estimation errors. Furthermore, it allows us to discern the main rigidly moving parts in the sequence by producing a scale-space of rigid motions. Visualization is accomplished by projecting the embedded matrix onto 3 different representative vectors in $\mathbb{R}^{12}$. The regularization is implemented using the CUDA framework, with computation times shown in Table 1. Using 15 outer iterations and 3 Gauss-Seidel iterations per inner iteration, practical convergence is achieved in 63 milliseconds on an NVIDIA GTX-580 card for QVGA-sized images, demonstrating the efficiency of our algorithm and its potential for real-time applications. This is especially important for applications such as gesture recognition where fast computation is important. A residual plot in the left sub-figure of Figure 6 demonstrates convergence of our method.

Outer iterations	15	15	25	50
GS iterations	1	3	1	1
320×240	49	63	81	160
640×480	196	250	319	648
1920×1080	1745	2100	2960	5732

Table 1. GPU processing times for various sizes of images, given in milliseconds.

Furthermore, since the main constraint for $SO(n)$ matrices (or the rotation part of $SE(n)$ matrices) is that of orthogonality, we measure during convergence

$$err_{\text{orth}}(u) = \left\| U^T U - I \right\|_F^2 \tag{60}$$

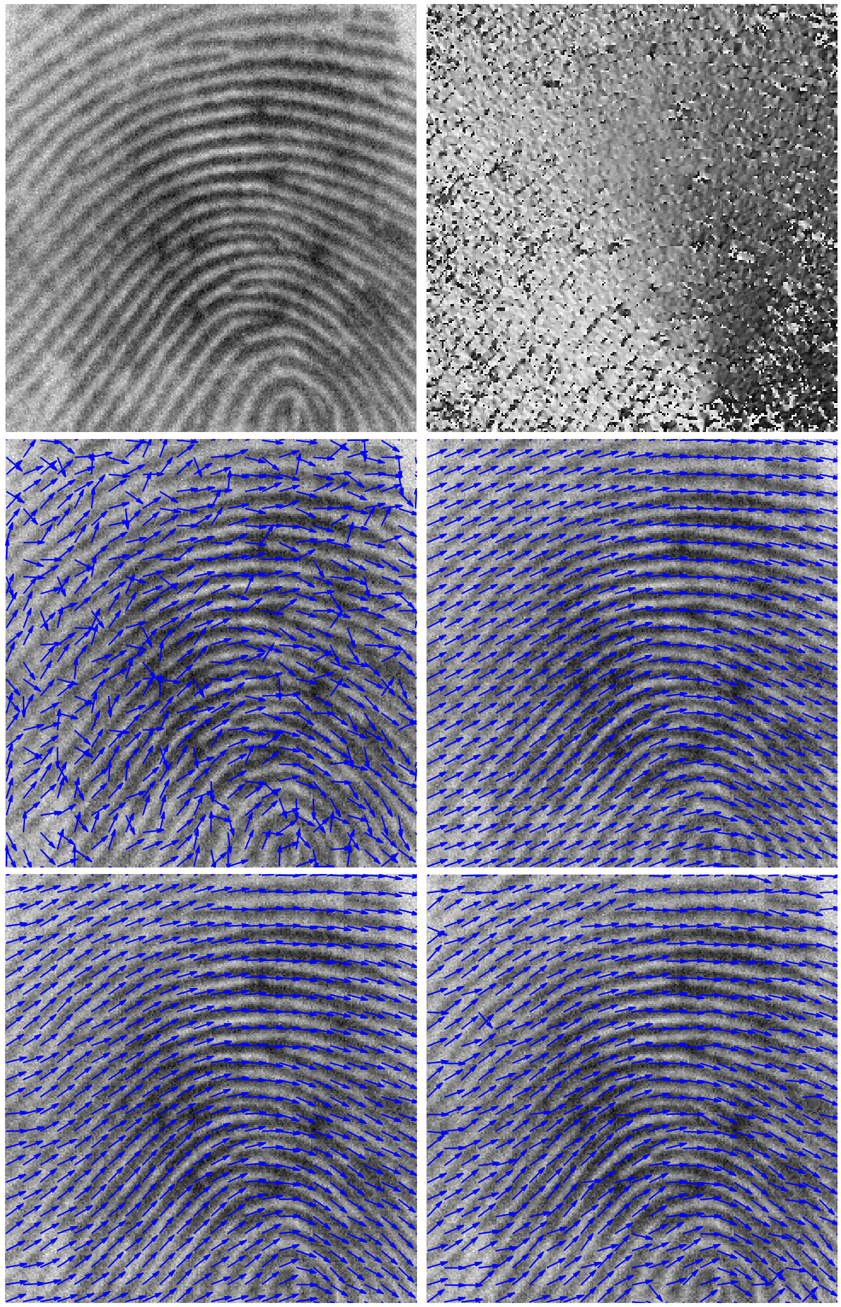

Fig. 4. TV regularization of $SO(2)$ data based on fingerprint direction estimation. Left-to-right, top-to-bottom: The fingerprint image with added Gaussian noise of $\sigma = 0.05$, the detected direction angles, the detected directions displayed as arrows, the detected directions after regularization with $\lambda = 3$, regularization results using Equation 9, regularization results based on higher-order diffusion term with $\lambda = 6$, the regularization result by Sochen et al. [46].

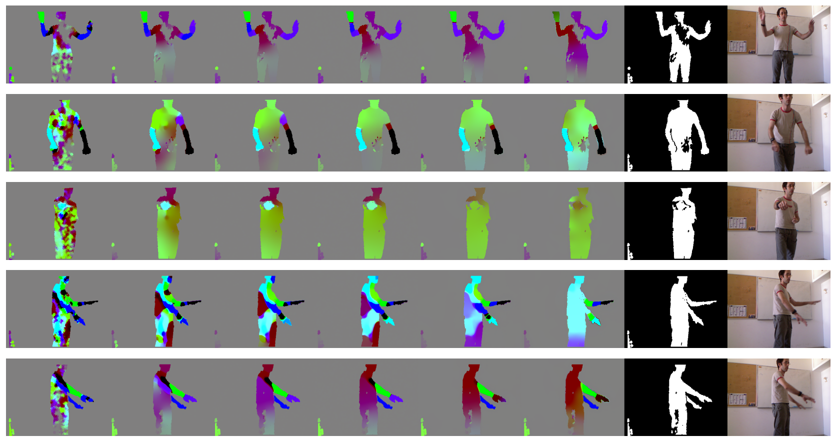

Fig. 5. Regularization of $SE(3)$ images obtained from local ICP matching of the surface patch between consecutive Kinect depth frames. Left-to-right: diffusion scale-space obtained by different values of λ: $1.5, 1.2, 0.7, 0.2, 0.1, 0.05$, the foreground segmentation based on the depth, and an intensity image of the scene.

The plot of err_{orth} as a function of the iterations is shown in the right sub-figure of Figure 6. The plot demonstrates the enforcement of the constraint $u \in \mathcal{G}$ by the augmented Lagrangian scheme for most of the convergence. The close adherence to the isometry assumption validates in practice our usage of the regularization proposed in Equation 7 for isometry groups.

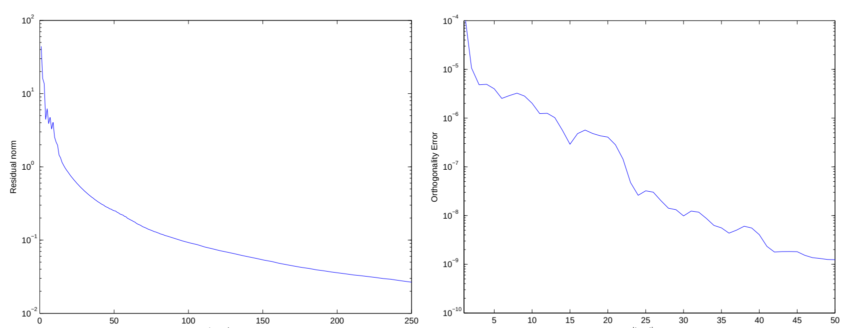

Fig. 6. A residual plot (left), and orthogonality error norm plot (right) for $SE(3)$ denoising as demonstrated in Figure 5, for $\lambda = 0.2$.

6.3 DTI regularization

In Figure 7 we demonstrate a smoothing of DTI data from [32], based on the scheme suggested in Section 3.5, using the Slicer3D tool in order to visualize the tensors via ellipsoid glyphs. Figure 8 demonstrates the convergence rate for the regularization. MSE of the matrix representation was 0.0406 in the corrupted image and 0.0248 in the regularized image.

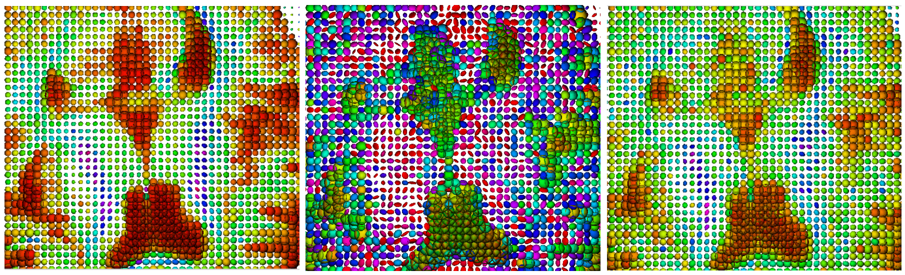

Fig. 7. TV denoising of images with diffusion tensor data. Left-to-right: the original image, an image with added component-wise Gaussian noise of $\sigma = 0.1$, and the denoised image with $\lambda = 30$.

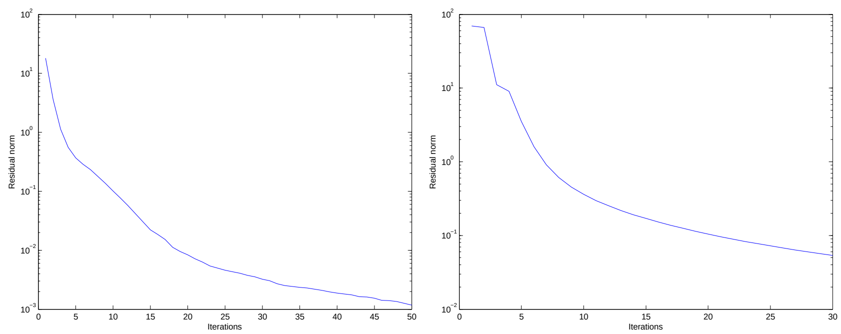

Fig. 8. A residual plot for DTI denoising (left) and reconstruction (right) as demonstrated in Figures 7,9, respectively.

In Figures 9,10 we demonstrate reconstruction of the DTI tensors, again based data from Lundervold et al. [32], using a set of 30 directional measurements. The measure ratios $\log\left(\frac{S_i}{S_0}\right)$ were added a Gaussian additive noise of standard deviation 100. The reconstructed image obtained by regularized reconstruction with $\lambda = 1 \times 10^{-3}$ had an MSE of 2.1×10^{-4}, compared to 8.9×10^{-3} without regularization.

7 Conclusions

In this chapter we demonstrate thed effectiveness of augmented Lagrangian regularization of matrix-valued maps. Specifically, we have shown the efficiency and effectiveness of the resulting total-variation regularization of images with matrix-valued data

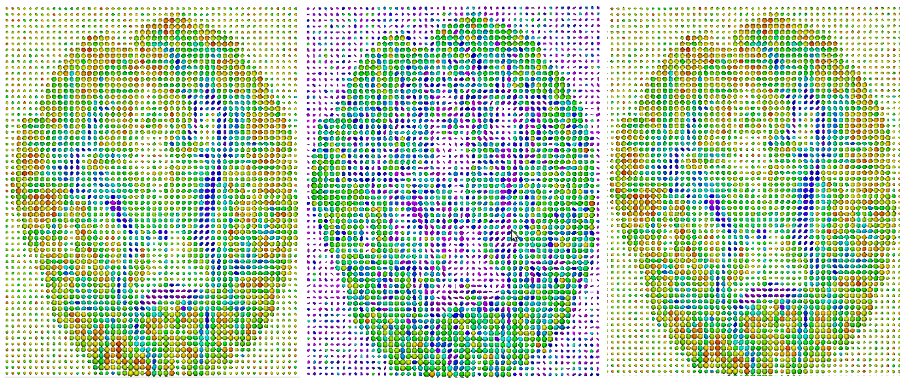

Fig. 9. TV-regularized reconstruction of images with diffusion tensor data. Left-to-right: the original image, an image with added component-wise Gaussian noise, and the denoised image. Noise was of standard deviation 100, $\lambda = 1 \times 10^{-3}$.

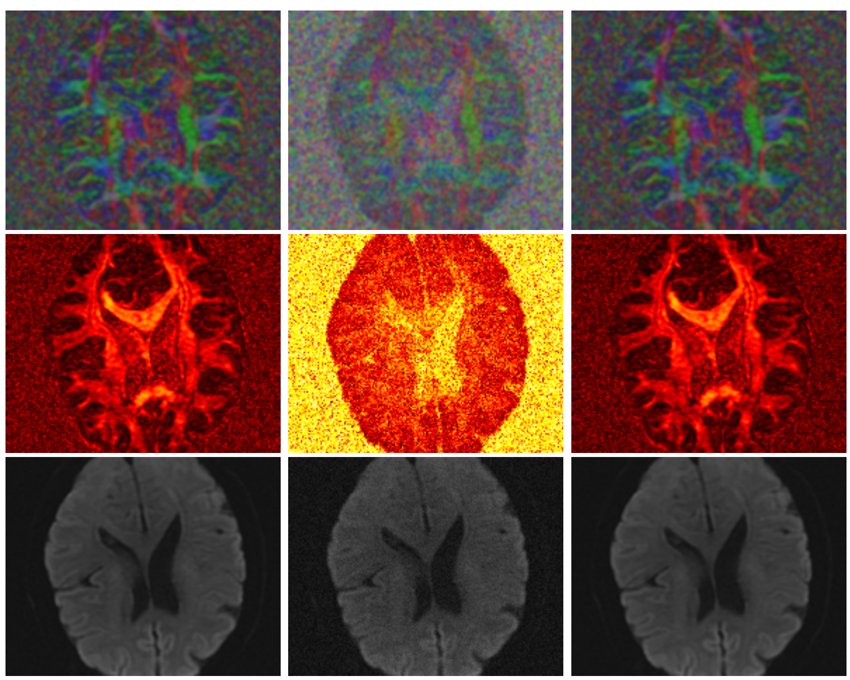

Fig. 10. TV-regularized reconstruction of diffusion tensor data. Left-to-right: the original reconstruction without noise, the noisy least-squares fitting solution (used as initialization), and the regularized reconstruction result. Top-to-bottom: a visualization of the principal directions, the fractional anisotropy, and the mean diffusivity. The noise added to the field ratio logarithm was of strength 100, $\lambda = 1 \times 10^{-3}$.

taken from $SO(n)$, $SE(n)$, and $SPD(n)$. For the case of $SPD(n)$ we have shown the method's usefulness for denoising and regularized reconstruction of DTI data, as well as noted the convexity of the resulting optimization problem.

In future work we intend to explore the various ways of handling the matrix-valued regularization problem and the coupling between matrix elements, as well as extend our work into different data types and applications.

References

1. Fingerprints Verification Competition database.
2. S. Ali and M. Shah. A Lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *Computer Vision and Pattern Recognition*, pages 1–6, 2007.
3. H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Lojasiewicz inequality. *Math. Oper. Res.*, 35:438–457, 2010.
4. P. J. Basser, J. Mattiello, and D. LeBihan. MR diffusion tensor spectroscopy and imaging. *Biophysical journal*, 66(1):259–267, Jan. 1994.
5. S. Basu, P. T. Fletcher, and R. T. Whitaker. Rician noise removal in diffusion tensor MRI. In R. Larsen, M. Nielsen, and J. Sporring, editors, *MICCAI (1)*, volume 4190 of *Lecture Notes in Computer Science*, pages 117–125. Springer, 2006.
6. E. Bayro-Corrochano and J. Ortegón-Aguilar. Lie algebra approach for tracking and 3D motion estimation using monocular vision. *Image Vision Comput.*, 25:907–921, June 2007.
7. Ø. Bergmann, O. Christiansen, J. Lie, and A. Lundervold. Shape-adaptive DCT for denoising of 3D scalar and tensor valued images. *J. Digital Imaging*, 22(3):297–308, 2009.
8. P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992.
9. S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
10. A. D. Bue, X. J. ao, L. Agapito, and M. Paladini. Bilinear factorization via augmented Lagrange multipliers. In *European Conf. Computer Vision*, pages 283–296. Springer-Verlag, 2010.
11. B. Burgeth, S. Didas, L. Florack, and J. Weickert. A generic approach to the filtering of matrix fields with singular PDEs. In *Scale Space and Variational Methods in Computer Vision*, pages 556–567, 2007.
12. E. Celledoni and B. Owren. Lie group methods for rigid body dynamics and time integration on manifolds. *Computer Methods in Applied Mechanics and Engineering*, 19:421–438, 1999.
13. B. Chen and E. W. Hsu. Noise removal in magnetic resonance diffusion tensor imaging. *Magnetic resonance in medicine : official journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine*, 54(2):393–401, Aug. 2005.
14. R. Deriche, D. Tschumperle, and C. Lenglet. DT-MRI estimation, regularization and fiber tractography. In *ISBI*, pages 9–12, 2004.
15. R. Duits and B. Burgeth. Scale spaces on Lie groups. In *Scale Space and Variational Methods in Computer Vision*, pages 300–312, Berlin, Heidelberg, 2007. Springer Verlag.
16. Y. Dumortier, I. Herlin, and A. Ducrot. 4D tensor voting motion segmentation for obstacle detection in autonomous guided vehicle. In *IEEE Int. Vehicles Symp.*, pages 379–384, 2008.
17. I. Ekeland and R. Temam. *Convex Analysis and Variational Problems*. CMS Books in Mathematics. SIAM, Philadelphia, 1999.
18. R. Fletcher. Semi-definite matrix constraints in optimization. *SIAM J. on Cont. and Optimization*, 23(4):493–513, 1985.

19. W. Gibson. On the least-squares orthogonalization of an oblique transformation. *Psychometrika*, 27:193–195, 1962.
20. Y. Gur and N. A. Sochen. Regularizing flows over lie groups. *Journal of Mathematics in Imaging and Vision*, 33(2):195–208, 2009.
21. B. C. Hall. *Lie Groups, Lie Algebras,and Representations, An Elementary Introduction*. Springer Verlag, 2004.
22. M. R. Hesteness. Multipliers and gradient methods. *Journal of Optimization Theory and Applications*, 4:303–320, 1969.
23. N. J. Higham. Matrix nearness problems and applications. In *Applications of Matrix Theory*, pages 1–27. Oxford University Press, Oxford, 1989.
24. A. Iserles, H. Z. Munthe-kaas, S. P. Nørsett, and A. Zanna. Lie group methods. *Acta Numerica*, pages 215–365, 2000.
25. R. Kimmel, , and N. Sochen. Orientation diffusion or how to comb a porcupine. *special issue on PDEs in Image Processing, Computer Vision, and Computer Graphics, Journal of Visual Communication and Image Representation*, 13:238–248, 2002.
26. N. Kiryati, T. Riklin-Raviv, Y. Ivanchenko, and S. Rochel. Real-time abnormal motion detection in surveillance video. In *ICPR*, pages 1–4, 2008.
27. C. Koay, J. Carew, A. Alexander, P. Basser, and M. Meyerand. Investigation of anomalous estimates of tensor-derived quantities in diffusion tensor imaging. *Magn Reson Med*, 2006.
28. M. Kobilarov, K. Crane, and M. Desbrun. Lie group integrators for animation and control of vehicles. *ACM Trans. Graph.*, 28(2):1–14, 2009.
29. P. M. Larochelle, A. P. Murray, and J. Angeles. *On advances in robot kinematics*, chapter SVD and PD based projection metrics on SE(N), pages 13–22. Kluwer, 2004.
30. C. Lenglet, J. S. W. Campbell, M. Descoteaux, G. Haro, P. Savadjiev, D. Wassermann, A. Anwander, R. Deriche, G. B. Pike, and G. Sapiro. Mathematical methods for diffusion MRI processing. *NeuroImage*, 45(1):S111–S122, Mar. 2009.
31. D. Lin, W. Grimson, and J. Fisher. Learning visual flows: A Lie algebraic approach. In *Computer Vision and Pattern Recognition*, pages 747–754, 2009.
32. A. Lundervold. On consciousness, resting state fMRI, and neurodynamics. *Nonlinear Biomed Phys*, 4 Suppl 1, 2010.
33. J. H. Manton. Optimization algorithms exploiting unitary constraints. *IEEE Trans. Signal Process.*, 50(3):635–650, March 2002.
34. R. Mehran, B. E. Moore, and M. Shah. A streakline representation of flow in crowded scenes. In *European Conf. Computer Vision*, 2010.
35. L. Moisan. Perspective invariant movie analysis for depth recovery. *Proc. SPIE*, 2567:84–94, 1995.
36. M. Nicolescu and G. Medioni. A voting-based computational framework for visual motion analysis and interpretation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27:739–752, May 2005.
37. F. C. Park, J. E. Bobrow, and S. R. Ploen. A Lie group formulation of robot dynamics. *Int. J. Rob. Res.*, 14:609–618, December 1995.
38. W. Park, Y. Liu, Y. Zhou, M. Moses, and G. s. Chirikjian. Kinematic state estimation and motion planning for stochastic nonholonomic systems using the exponential map. *Robotica*, 26:419–434, July 2008.
39. X. Pennec, P. Fillard, and N. Ayache. A Riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006.
40. P. Perona. Orientation diffusions. *IEEE Trans. Image Process.*, 7(3):457–467, 1998.
41. M. J. Powell. *Optimization*, chapter A method for nonlinear constraints in minimization problems, pages 283–298. Academic Press, 1969.
42. I. U. Rahman, I. Drori, V. C. Stodden, D. L. Donoho, and P. Schroeder. Multiscale representations of manifold-valued data. Technical report, Stanford, 2005.

43. M. Raptis and S. Soatto. Tracklet descriptors for action modeling and video analysis. In *European Conf. Computer Vision*, pages 577–590, Sep. 2010.
44. G. Rosman, A. M. Bronstein, M. M. Bronstein, A. Wolf, and R. Kimmel. Group-valued regularization for motion segmentation of dynamic non-rigid shapes. In *SSVM*, 2011.
45. L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D Letters*, 60:259–268, 1992.
46. C. Sagiv, N. A. Sochen, and R. Kimmel. Stereographic combing a porcupine or studies on direction diffusion in image processing. *SIAM J. Appl. Math.*, 64(5):1477–1508, 2004.
47. R. Salvador, A. Pena, D. K. Menon., T. Carpenter, J. Pickard, and E. Bullmore. Formal characterization and extension of the linearized diffusion tensor model. *Human brain mapping*, 24(2):144–155, Feb. 2005.
48. G. Steidl, S. Setzer, B. Popilka, and B. Burgeth. Restoration of matrix fields by second-order cone programming. *Computing*, 81(2-3):161–178, 2007.
49. E. O. Stejskal and J. E. Tanner. Spin diffusion measurements: Spin echoes in the presence of a time-dependent field gradient. *Journal of Chemical Physics*, 42:288–292, 1965.
50. J. Stillwell. *Naive Lie Theory*. Undergraduate texts in mathematics. Springer-Verlag New York, 2008.
51. X.-C. Tai and C. Wu. Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model. In *Scale Space and Variational Methods in Computer Vision*, pages 502–513, 2009.
52. B. Tang, G. Sapiro, and V. Caselles. Diffusion of general data on non-flat manifolds viaharmonic maps theory: The direction diffusion case. *International Journal of Computer Vision*, 36:149–161, February 2000.
53. D. Tschumperlé and R. Deriche. Vector-valued image regularization with pdes: A common framework for different applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:506–517, 2005.
54. P. Tseng. Coordinate ascent for maximizing nondifferentiable concave functions. LIDS-P 1940, MIT, 1988.
55. P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:2273–2286, 2011.
56. O. Tuzel, F. Porikli, and P. Meer. Learning on lie groups for invariant detection and tracking. In *Computer Vision and Pattern Recognition*, 2008.
57. B. C. Vemuri, Y. Chen, M. Rao, T. McGraw, Z. Wang, and T. Mareci. Fiber tract mapping from diffusion tensor MRI. In *Proceedings of the International Conference on Variational, Geometry and Level Sets Methods in Computer Vision*, pages 81–88. IEEE Computer Society, 2001.
58. L. A. Vese and S. J. Osher. Numerical methods for p-Harmonic flows and applications to image processing. *SIAM Journal on Numerical Analysis*, 40(6):2085–2104, 2002.
59. M. Žefran, V. Kumar, and C. Croke. On the generation of smooth three-dimensional rigid body motions. *Robotics and Automation, IEEE Transactions on*, 14(4):576 –589, Aug. 1998.
60. M. Žefran, V. Kumar, and C. Croke. Metrics and connections for rigid-body kinematics. *I. J. Robotic Res.*, 18(2):242, 1999.
61. Y. Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM J. Imag. Sci.*, 1(3):248–272, 2008.
62. J. Weickert and T. Brox. Diffusion and regularization of vector- and matrix-valued images. volume 313 of *Inverse problems, image analysis, and medical imaging*, 2002.
63. Z. Wen, D. Goldfarb, and W. Yin. Alternating direction augmented Lagrangian methods for semidefinite programming. CAAM TR09-42, Rice university, 2009.
64. C.-F. Westin, S. Peled, H. Gudbjartsson, R. Kikinis, and F. A. Jolesz. Geometrical diffusion measures for MRI from tensor basis analysis. In *ISMRM*, 1997.

65. N. Wiest-Daesslé, S. Prima, P. Coupé, S. P. Morrissey, and C. Barillot. Non-local means variants for denoising of diffusion-weighted and diffusion tensor MRI. pages 344–351, Berlin, Heidelberg, 2007. Springer-Verlag.
66. C. Wu and X.-C. Tai. Augmented lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models. *SIAM J. Imaging Sciences*, 3(3):300–339, 2010.
67. C. Wu, J. Zhang, and X. Tai. Augmented lagrangian method for total variation restoration with non-quadratic fidelity. *Inverse Problems and Imaging*, 5(1):237–261, 2011.