METHODS AND APPLICATIONS OF ANALYSIS.
Vol. 21, No. 1, pp. 105–122, March 2014

005

AUGMENTED-LAGRANGIAN REGULARIZATION OF MATRIX-VALUED MAPS*

GUY ROSMAN[§†], XUE-CHENG TAI[‡], RON KIMMEL[§], AND ALFRED M. BRUCKSTEIN[§]

Abstract. We propose a novel framework for fast regularization of matrix-valued images. The resulting algorithms allow a unified treatment for a broad set of matrix groups and manifolds. Using an augmented-Lagrangian technique, we formulate a fast and highly parallel algorithm for matrix-valued image regularization.

We demonstrate the applicability of the framework for various problems, such as motion analysis and diffusion tensor image reconstruction, show the formulation of the algorithm in terms of split-Bregman iterations and discuss the convergence properties of the proposed algorithms.

Key words. Regularization, Lie-groups, total-variation, split-Bregman, matrix-manifolds, diffusion-imaging, rotations, articulated motion.

AMS subject classifications. 65K10, 58JXX.

1. Introduction.

Matrix-manifolds and Matrix-valued images have become an integral part of computer vision and image processing. Matrix-manifolds and groups have been used for tracking [43, 56], robotics [38, 58, 10], motion analysis [45, 20], image processing and computer vision [9, 40, 42, 45, 60], as well as medical imaging [4, 39]. Efficient regularization of matrix-valued images is therefore highly important in the fields of image analysis and computer vision. This includes applications such as direction diffusion [29, 53, 59] and scene motion analysis [33] in computer vision, as well as diffusion tensor MRI (DT-MRI) regularization [5, 15, 25, 50, 54] in medical imaging.

We present an augmented Lagrangian method for efficient regularization of matrix-valued images, or maps. We assume the matrix-manifold to have an efficient projection operator onto it from some embedding into a Euclidean space, and that the distortion associated with this mapping is not too large in term of the metric accompanying these spaces.

Examples of matrix-manifolds that are of interest include the special-orthogonal and special-Euclidean Lie-groups and the symmetric positive-definite matrices, as well as Stiefel manifolds. We show that the augmented Lagrangian technique allows us to separate the optimization process into a regularization update step of a map onto an embedding-space, and a per-pixel projection step. An efficient regularization step is shown for the *total-variation* (TV, [48]) regularization, and a second-order regularization penalizing the Hessian norm. Both the regularization step and the projection steps are simple to compute, fast and easily parallelizable using consumer graphic processing units (GPUs), achieving real-time processing rates. The resulting framework unifies algorithms using in several application domains into one framework, since they differ only in the choice of projection operator. While such an optimization problem could have been approached by general saddle-point solvers such as [12], the

*Received January 9, 2013; accepted for publication October 2, 2013.

†This research was supported by Israel Science Foundation grant no.1551/09 and by the European Community's FP7- ERC program, grant agreement no. 267414.

‡Department of Mathematics, University of Bergen, Johaness Brunsgate 12, Bergen 5007 Norway (tai@mi.uib.no).

§Computer Science Department, Technion 32000, Haifa, Israel ({rosman; ron; freddy}@ cs.technion.ac.il).

domain of our problem is not convex, requiring such algorithms to be revisited in order to demonstrate convergent behavior.

In order to obtain fast update steps we add two auxiliary fields, with appropriate constraints. One field approximates the gradient of the image and simplifies the total-variation cost function minimization, as done, for example, in [11, 37, 23, 52]. Another field approximates the image, but is forced during its update to stay on matrix manifold, turning the group constraint into a simple projection operator. This results in a unified framework for processing of $SO(n)$, $SE(n)$ and $SPD(n)$ images, as we describe in Section 3. This framework was initially presented in a recent conference paper [47] and we now expand upon it, with additional explanations and a partial convergence proof. In addition, we relate in this section the proposed algorithms to split-Bregman iterations and describe their convergence properties. In Section 4 we demonstrate a few results of our method, in regularization of 3D motion analysis, medical image analysis, and direction diffusion. Section 5 concludes the paper.

2. A short introduction to Lie-groups and matrix manifolds. We now shortly describe the matrix manifolds we deal with in our algorithm. Their structure allows us to define priors on matrix-valued data in computer vision and has been the subject of intense research efforts, especially involving statistics of matrix-valued data [39], optimization on matrix-manifolds [18], and regularization of matrix-valued images [54], as well as describing the dynamics of spatial processes involving *Lie-group* data [33]. Lie-groups are algebraic groups endowed with a differentiable manifold structure and an appropriate group action. We briefly describe the Lie-groups our algorithm deals with, and refer the reader to the literature for an introduction to Lie-groups [26].

The rotations group $SO(n)$ - The group $SO(n)$ describes all rotation matrices of the n-dimensional Euclidean space,

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

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 the rotations manifold $SO(n)$ and the manifold $\mathbb{R}^n$ representing all translations of the Euclidean space. In matrix form this group is written as

$$SE(n) = \left\{ \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix}, \mathbf{R} \in SO(n), \mathbf{t} \in \mathbb{R}^n \right\}. \tag{2}$$

The symmetric positive definite set $SPD(n)$ - Another matrix manifold that has an efficient projection operator is the cone of symmetric positive definite matrices. This matrix manifold has been studied extensively in control theory (see [19] for example), 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 set is given in matrix form as

$$SPD(n) = \{ \mathbf{A} \in \mathbb{R}_{n\times n}, \mathbf{A} \succeq 0 \}. \tag{3}$$

3. An augmented Lagrangian regularization algorithm for matrix-valued images. We now proceed to describe a fast regularization algorithm for images with matrix-valued data, referred to as Algorithm 1. The standard regularization problem of Lie-groups maps is formulated in terms of the Lie-algebra,

$$\underset{u\in\mathcal{G}}{\operatorname{argmin}} \int \|u^{-1}\nabla u\| + \lambda\|u - u_0\|^2 dx, \tag{4}$$

where $\|\cdot\|$ is the Frobenius norm, u represents an element in an embedding of the Lie-group $\mathcal{G}$ into Euclidean space. We use the notation ∇u to denote the Jacobian of u, described as a column-stacked vector. We note that we use the same notation to represent the Lie-group element, its matrix representation, and the embedding into Euclidean 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. This formulation parallels the time derivatives in construction of Lie-group integrators [10], and defines smoothness in the tangent plane of the Lie-group. Smoothness expressed in the regularization term $\|u^{-1}\nabla u\|$ is in sense of the geometry of the Lie-group, via the Lie-algebra, but this may not generalize to other matrix groups. Furthermore, its minimization is inefficient as it requires computing the logarithm and exponential maps at each pixel update. In order to obtain a fast regularization scheme that applies for all matrix groups, we look instead at the regularity of an embedding of the Lie-group into Euclidean space,

$$\underset{u\,\in\,\mathcal{G}}{\operatorname{argmin}} \int \|\nabla u\| + \lambda\|u - u_0\|^2 dx, \tag{5}$$

where $\|\nabla u\|$ denotes (by abuse of notation) the Frobenius norm of the Jacobian of the map from the domain ($\mathbb{R}^2$ or $\mathbb{R}^3$) into the embedding space. This allows us to consider also matrix manifolds that are not Lie-groups, such as symmetric positive-definite matrices and Stiefel matrices, as part of the same framework. In our formulation, elements of $SO(n)$ can be embedded into $\mathbb{R}^m, m = n^2$, and elements of $SE(n)$ can similarly be embedded into $\mathbb{R}^m, m = n(n+1)$. The elements of $SPD(n)$ can be embedded into $\mathbb{R}^m, m = n(n+1)/2$.

The rationale behind the different regularization term $\|\nabla u\|$ stems from the fact that $SO(n)$ and $SE(n)$ are isometries of Euclidean space. Hence, the two regularization are the same for these sets of matrices. In general, such a regularization can always be defined, and can be related to the regularization term shown in Equation 4 whenever the data consists of nonsingular matrices. This regularization term has also been used for SPD matrices [57]. We refer the reader to our technical report [46] for a more in-depth discussion of this important point. 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 $\mathcal{G}$, where the equality constraint is enforced via augmented Lagrangian terms [27, 41]. The suggested augmented Lagrangian optimization now reads

$$\begin{aligned} &\min_{v\in\mathcal{G},u\in\mathbb{R}^m} \max_{\mu} \mathcal{L}(u,v;\mu) = \\ &\min_{v\in\mathcal{G},u\in\mathbb{R}^m} \max_{\mu} \int \begin{bmatrix} \|\nabla u\| + \lambda\|u-u_0\|^2 + \\ \frac{r}{2}\|u-v\|^2 + \operatorname{tr}(\mu^T(u-v)) \end{bmatrix} dx. \end{aligned} \tag{6}$$

Given a fixed Lagrange multiplier μ, the minimization w.r.t. u, v can be split into alternating minimization steps with respect to u and v, both of which lend themselves to an efficient and parallel optimization. Specifically, we can further reformulate the regularization of u in the same way as Wu and Tai [61], by introducing an auxiliary variable p

$$\begin{aligned} &\min_{v\in\mathcal{G},u\in\mathbb{R}^m,p\in\mathbb{R}^{mn}} \max_{\mu} \mathcal{L}(u,v,p;\mu,\mu_2) = \\ &\min_{v\in\mathcal{G},u\in\mathbb{R}^m,p\in\mathbb{R}^{mn}} \max_{\mu} \int \begin{bmatrix} \|p\| + \lambda\|u-u_0\|^2 + \\ \frac{r}{2}\|u-v\|^2 + \operatorname{tr}(\mu^T(u-v)) + \\ \frac{r_2}{2}\|\nabla u - p\|^2 + \operatorname{tr}(\mu_2^T(\nabla u - p)) \end{bmatrix} dx, \end{aligned} \tag{7}$$

where μ_2 is the Lagrange multiplier associated with the constraint $p = \nabla u$, and r_2 is the related penalty coefficient.

3.1. Minimization w.r.t. v. The advantage of adding the auxiliary variable v is that minimization w.r.t v becomes a simple projection problem per pixel,

$$\begin{aligned} &\operatorname*{argmin}_{v\in\mathcal{G}} \frac{r}{2}\|v-u\|^2 + \operatorname{tr}(\mu^T(u-v)) \\ &= \operatorname*{argmin}_{v\in\mathcal{G}} \frac{r}{2}\left\|v-\left(\frac{\mu}{r}+u\right)\right\|^2 \\ &= \operatorname*{Proj}_{\mathcal{G}}\left(\frac{\mu}{r}+u\right), \end{aligned} \tag{8}$$

where $\operatorname{Proj}_{\mathcal{G}}$ denotes a projection operator onto the specific matrix-group $\mathcal{G}$. The numerical update step for $SO(n)$, $SE(n)$ and $SPD(n)$ will be explicitly given later on.

3.2. Minimization w.r.t. u. The update step w.r.t u in Equation 6 is a vectorial TV denoising problem

$$\operatorname*{argmin}_{u\in\mathbb{R}^m} \int \|\nabla u\| + \tilde{\lambda}\left\|u-\tilde{u}\left(u_0,v,\mu,r\right)\right\|^2 dx, \tag{9}$$

with $\tilde{u} = \frac{(2\lambda u_0 + r v + \mu)}{(2\lambda + r)}$. This problem can be solved via fast minimization techniques for TV regularization of vectorial images, such as [8, 17, 22]. In our case, we regularize the image using the algorithm [52], 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

$$\begin{matrix} \min \\ u\in\mathbb{R}^m \\ p\in\mathbb{R}^{2m} \end{matrix} \max_{\mu_2} \int \begin{bmatrix} \tilde{\lambda}\left\|u-\tilde{u}\left(u_0,v,\mu,r\right)\right\|^2 + \|p\| \\ +\mu_2^T(p-\nabla u) + \frac{r_2}{2}\|p-\nabla u\|^2 \end{bmatrix} dx. \tag{10}$$

We solve for u using the Euler-Lagrange equation,

$$2\tilde{\lambda}(u-\tilde{u}) + (\operatorname{div}\mu_2 + r_2 \operatorname{div} p) + \Delta u = 0, \tag{11}$$

for example, in the Fourier domain, or by Gauss-Seidel iterations. We have chosen Gauss-Seidel iterations since complete minimization of the functional is not required at each substep.

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

$$\operatorname*{argmin}_{p\in\mathbb{R}^{2m}} \int \|p\| + \mu_2^T p + \frac{r_2}{2}\|p-\nabla u\|^2, \tag{12}$$

with the closed-form solution [52]

$$p = \frac{1}{r_2}\max\left(1-\frac{1}{\|w\|},0\right) w, w = r_2\nabla u - \mu_2. \tag{13}$$

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

$$\begin{aligned} \mu^k &= \mu^{k-1} + r\left(v^k - u^k\right), \\ \mu_2^k &= \mu_2^{k-1} + r_2\left(p^k - \nabla u^k\right). \end{aligned} \tag{14}$$

An algorithmic description is summarized as Algorithm 1.

Algorithm 1 Fast TV regularization of matrix-valued data

1: **for** $k = 1, 2, \ldots,$ until convergence **do**
2: Update $u^k(x)$, according to Equation (11). For the modified scheme as discussed in Sub-subsection 3.4.1, use Equation 21.
3: Update $p^k(x)$, according to Equation (13).
4: Update $v^k(x)$, by projection onto the matrix group,
- For $SO(n)$ matrices, according to Equation (19). For the modified scheme use Equation 22.
- For $SE(n)$ matrices, according to Equation (28). See Sub-subsection 3.4.1 for details on how to modify the update step.
- For $SPD(n)$ matrices, according to Equation (29). See Sub-subsection 3.4.1 for details on how to modify the update step.

5: Update $\mu^k(x), \mu_2^k(x)$, according to Equation (14).
6: **end for**

3.3. Split-Bregman method for matrix-valued regularization. An additional interpretation of augmented-Lagrangian total variation regularization has been suggested in [52], relating it to split-Bregman techniques, and specifically, the method suggested in [31]. In fact, for the case of $\mathcal{G} = S^m$, the projection operator for v would be the same as the one used in [31]. In split-Bregman iterations [24], in each iteration a *Bregman distance* [7] is minimized.

Looking at the function

$$E(p, u, v) = \int \|p\| + \|u - u_0\|^2, \tag{15}$$

and vector sequence

$$d_{BREG}^k = \left(d_u^k, d_p^k, d_v^k\right) = -\left(\operatorname{div} \mu_2^k + \mu^k, \mu_2^k, -\mu^k\right), \tag{16}$$

we use the Bregman distance

$$\begin{aligned} &D^{d_{BREG}^k}\left((u, q, v), \left(u^k, p^k, v^k\right)\right) = \\ &\int \|p\| + \|u - u_0\|^2 - \|p^k\| - \left\|u^k - u_0\right\|^2 - \left\langle d_{BREG}^k, (u, p, v)\right\rangle. \end{aligned} \tag{17}$$

Re-examining our inner update step in algorithm 1, we can reformulate our algorithm (between updates of the Lagrange multipliers) in the form of a split-Bregman

iterations

$$
\begin{aligned}
(18)\qquad & \left(u^{k+1}, p^{k+1}, v^{k+1}\right) = \\
& \underset{u,p,v}{\operatorname{argmin}} \; D^{\left(d_u^k, d_p^k, d_v^k\right)}\left((u,p,v), \left(u^k, p^k, v^k\right)\right) \\
& \quad + \tfrac{r}{2}\int \|p - \nabla u\|^2 + \tfrac{r_2}{2}\int \|u - v\|^2 \quad = \\
& \underset{u,p,v}{\operatorname{argmin}} \int \; \|p\| + \|u - u_0\|^2 + \left\langle \operatorname{div} \mu_2^k + \mu^k, u \right\rangle \\
& \quad + \left\langle \mu^k, p \right\rangle + \left\langle -\mu_2^k, v \right\rangle + \tfrac{r_2}{2}\|p - \nabla u\|^2 + \tfrac{r}{2}\|u - v\|^2 \quad = \\
& \underset{u,p,v}{\operatorname{argmin}} \int \; \|p\| + \|u - u_0\|^2 + \left\langle \mu_2^k, p - \nabla u \right\rangle \\
& \quad + \left\langle \mu^k, u - v \right\rangle + \tfrac{r_2}{2}\|p - \nabla u\|^2 + \tfrac{r}{2}\|u - v\|^2 \quad .
\end{aligned}
$$

We note that we take into account the nature of v while minimizing the Bregman distance only. In this sense, and in the choice of vector d_{BREG}^k the algorithm differs from the split-Bregman method. This is not surprising as our domain is not necessarily convex, and the convergence properties of the split-Bregman iteration cannot automatically hold for non-convex domains. Partial convergence proofs for the algorithm can be obtained in the context of the augmented-Lagrangian formulation, with some small modification, as mentioned in the following section for non-convex matrix manifolds.

3.4. 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 [21]. 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

$$
\begin{aligned}
(19)\qquad & v^{k+1} = \underset{SO(n)}{\operatorname{Proj}} \left(\frac{\mu}{r} + u^k\right) = \mathbf{U}(x)\mathbf{V}^T(x), \\
& \mathbf{USV}^T = \left(\frac{\mu}{r} + u^k\right).
\end{aligned}
$$

Other possibilities include using the Euler-Rodrigues formula, quaternions, or the polar decomposition [32]. We note that the non-convex domain $SO(n)$ makes the analysis of global convergence for this scheme quite elaborate.

3.4.1. Convergent behavior of the inner iterations. Despite the non-convex domain, and non-continuous projection operator, the inner iterations (update of u, v, p) of a modified variant of the algorithm, in the case of $\mathcal{G} = SO(n)$ (and $\mathcal{G} = SE(n)$) can be easily made convergent in a weak sense using the method inspired by the work of Attouch et al. [3], as described in our technical report [46]. In this limited setup, we hold the Lagrange multipliers μ, μ_2 fixed, and update p, u, v, minimizing them as described above. Adapting our notation to that of Attouch et al. [3], we rewrite

$$
\begin{aligned}
(20)\qquad & f(u,p) = \|p\| + \mu_2^T (p - \nabla u) + \frac{r_2}{2}\|p - \nabla u\|^2, \\
& g(v) = i_g(v), \\
& Q(u,p,v) = \mu^T (u - v) + \frac{r}{2}\|u - v\|^2, \\
& L(u,v,p) = f(u,p) + Q(u,p,v) + g(v),
\end{aligned}
$$

where Q, f, g would fill similar roles as in [3], and $i_g(v)$ is the indicator function for the group g. We note that care needs to be taken when adapting the proofs from [3],

as in our case the minimization step of v is confined to a subset of the Euclidean space which is not an affine subspace, and hence many of the proofs utilizing the subgradient w.r.t v need to be significantly revised. We can still, however, prove convergence of the residuals towards zero in the following manner. We modify our update steps to be

$$u^k = \underset{u}{\operatorname{argmin}}\, L(u, p^{k-1}, v^{k-1}) + \frac{1}{2\theta}\|u - u^{k-1}\|^2, \tag{21}$$

$$v^k = \underset{v}{\operatorname{argmin}}\, L(u^k, p^{k-1}, v) + \frac{1}{2\theta}\|v - v^{k-1}\|^2, \tag{22}$$

$$p^k = \underset{p}{\operatorname{argmin}}\, L(u^k, p, v^k), \tag{23}$$

where θ is a (finite and positive) constant coefficient. We begin by noting according to Equation 23 that

$$L\left(u^k, p^k, v^k\right) \leq L\left(u^k, p^{k-1}, v^k\right). \tag{24}$$

Using Equation 22 we get

$$L\left(u^k, p^{k-1}, v^k\right) + \frac{1}{2\theta}\|v^k - v^{k-1}\|^2 \leq L\left(u^k, p^{k-1}, v^{k-1}\right).$$

Finally, using Equation 21 we get

$$L\left(u^k, p^{k-1}, v^{k-1}\right) + \frac{1}{2\theta}\|u^k - u^{k-1}\|^2 \leq L\left(u^{k-1}, p^{k-1}, v^{k-1}\right),$$

resulting in the inequality

$$L\left(u^k, p^k, v^k\right) + \frac{1}{2\theta}\|u^k - u^{k-1}\|^2 + \frac{1}{2\theta}\|v^k - v^{k-1}\|^2 \leq L\left(u^{k-1}, p^{k-1}, v^{k-1}\right),$$

or alternatively

$$\|u^k - u^{k-1}\|^2 + \|v^k - v^{k-1}\|^2 \leq 2\theta\left(L\left(u^{k-1}, p^{k-1}, v^{k-1}\right) - L\left(u^k, p^k, v^k\right)\right). \tag{25}$$

Since the sequence of $L\left(u^k, p^k, v^k\right)$ is non-increasing, and is bounded from below (because of the quadratic structure of L with respect to u, v, p) for a given set of multipliers, we know that

$$2\theta\left(L\left(u^{k-1}, p^{k-1}, v^{k-1}\right) - L\left(u^k, p^k, v^k\right)\right) \to 0, \tag{26}$$

and therefore

$$\|u^k - u^{k-1}\|^2 + \|v^k - v^{k-1}\|^2 \to 0. \tag{27}$$

Thus, we can show that the residual decreases towards 0, and give some assurance as to the convergent behavior of the algorithm. A complete convergence analysis is not straightforward, as mentioned in [31]. Empirical results seem to demonstrate strong convergence properties in a variety of applications and scenarios, for a wide variety of θ values. Convergence plots for a range of θ values is shown in Figure 1. As can be seen, there is a slight advantage in terms of convergence speed for high θ values and weak coupling, but in general, the method works well for a large variety of θ values.

Moreover, the case where only partial updates of u, v, p are performed is more elaborate and its analysis is left as future work. We note that for the case of total-variation regularization, properties of the split-Bregman iterations with partial accuracy have been analyzed by Yin and Osher [62].

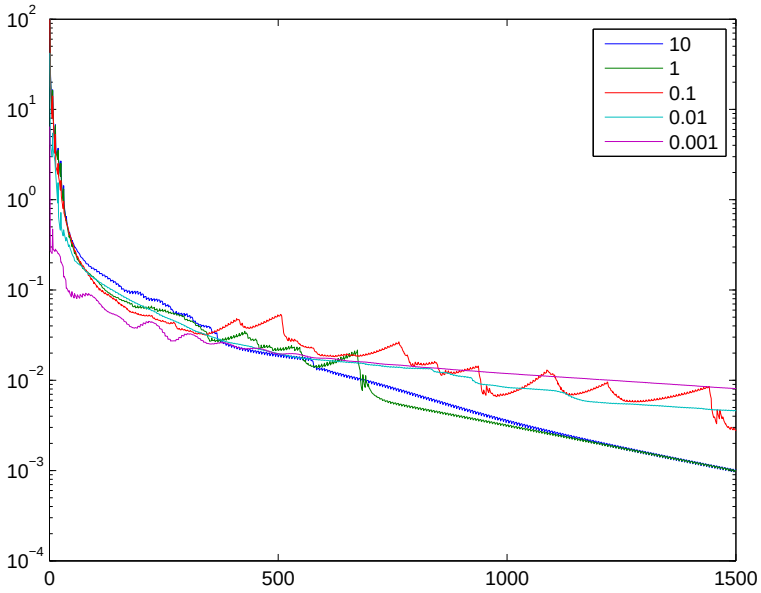

FIG. 1. *Residual plots for a variety of θ values, for the optimization problem given in Figure 2*

3.5. Regularization of maps onto $SE(n)$. In order to regularize images with values in $SE(n)$, we use the embedding of $SE(n)$ 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, with a similar partition for the Lagrange multipliers $\mu = (\mu_R, \mu_t)$. Updating v in line 4 of Algorithm 1 assumes the form

$$v_R^{k+1} = \underset{SO(n)}{\text{Proj}} \left(\frac{\mu_R}{r} + u_R^k \right), \; v_t^{k+1} = \left(\frac{\mu_t}{r} + u_t^k \right) \tag{28}$$

$$v^{k+1} = \underset{SE(n)}{\text{Proj}} (v^k) = (v_R^{k+1}, v_t^{k+1}).$$

Modification of the update scheme in order to obtain weak convergence guarantees can be done in a manner parallel to that shown in Subsection 3.4.

3.6. Regularization of maps onto $SPD(n)$. The technique described above can be used also for regularizing symmetric positive-definite matrices. Here, the intuitive choice of projecting the eigenvalues of the matrices onto the positive half-space is shown to be optimal [28]. Many papers dealing with the analysis of DT-MRI rely on the eigenvalue decomposition of the tensor as well, i.e. for tractography, anisotropy measurements, and so forth.

For $\mathcal{G} = SPD(n)$, the minimization problem w.r.t. v in step 3 of Algorithm 1 can be solved by projection of eigenvalues. Let $\mathbf{U} \operatorname{diag}(\boldsymbol{\lambda}) \mathbf{U}^T$ be the eigenvalue

decomposition of the matrix $\frac{\mu}{r} + u^k$. v is updated according to

$$v^{k+1} = \underset{SPD(n)}{\text{Proj}} \left(v^k\right) = \mathbf{U}(x) \operatorname{diag}\left(\hat{\boldsymbol{\lambda}}\right) \mathbf{U}^T(x), \tag{29}$$

$$\mathbf{U} \operatorname{diag}\left(\boldsymbol{\lambda}\right) \mathbf{U}^T = \left(\frac{\mu}{r} + u^k\right), \left(\hat{\boldsymbol{\lambda}}\right)_i = \max\left(\left(\boldsymbol{\lambda}\right)_i, 0\right),$$

where the matrix U is a unitary one, representing the eigenvectors of the matrix, and the eigenvalues $\left(\hat{\boldsymbol{\lambda}}\right)_i$ 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 [55], even without the modifications described in subsection 3.4. We refer the interested reader to our technical report [46]. An example of using the proposed method for DT-MRI denoising is shown in Section 4. In the example we have used the same scheme as in subsection 3.4 in order to use consistently the same algorithmic framework, although both in theory and in practice this was not necessary in order to assure convergence.

3.7. A higher-order prior for group-valued images. We note that the scheme we describe is susceptible to the staircasing effect, since it minimizes a total variation regularization of the map u. Several higher-order priors can be incorporated into our scheme, that avoid this effect. One such possible higher-order term generalizes the scheme presented by Wu and Tai [61], by replacing the per-element gradient operator with a Hessian operator. The resulting saddle-point problem becomes

$$\min_{\substack{u \in \mathbb{R}^m \\ p \in \mathbb{R}^{4m}, \\ v \in \mathcal{G}}} \max_{\mu_2} \int \begin{bmatrix} \|p\| + \tilde{\lambda} \left\|u - \tilde{u}\left(u_0, v, \mu, r\right)\right\|^2 \\ +\mu_2^T (p - H(u)) + \frac{r_2}{2} \|p - H(u)\|^2 \end{bmatrix} dx, \tag{30}$$

where H denotes the per-element Hessian operator,

$$\left(H(u)\right)_{i,\cdot} = \left(D_{xx}^{-+} u_i, D_{xy}^{++} u_i, D_{yx}^{++} u_i, D_{yy}^{-+} u_i\right), \tag{31}$$

where we use, for example D_{xx}^{-+} to describe the second order derivative obtained by first applying the forward and then the backward first order derivative. Minimizing this functional with respect to p is done by shrinkage, as described in [61]. Solving with respect to u given p is done by solving the resulting optimality system of equations [61],

$$2\tilde{\lambda}\left(u\right) + r_2 H^*\left(H(u)\right) = H^*(\mu_2) + r_2 H^*\left(p\right) + 2\tilde{\lambda}\tilde{u}, \tag{32}$$

where $H^*(\cdot)$ denotes the adjoint operator for the operator $H(\cdot)$,

$$H^*(p) = D_{xx}^{+-} p^1 + D_{xy}^{--} p^2 + D_{yx}^{--} p^3 + D_{yy}^{+-} p^4, \tag{33}$$

where p^i denotes the ith element of the per-pixel vector p for each element in u, using scalar notations in order to avoid further complicating the notation. We refer the reader to [61] for the complete discussion and definition of these operators, and remark that in our case, Gauss-Seidel iteration were used instead of a Fourier-domain solver.

We show an example using the appropriately modified scheme, for the case of $\mathcal{G} = SO(2)$ in Figures 2, 3.

4. Numerical results. As discussed above, the proposed algorithmic framework is considerable general and suitable for various applications. We now show several examples from different applications domain to demonstrate our algorithm.

4.1. 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, 30, and references therein], vehicle control [16], crowd behavior analysis [35], 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 obtain a smoothed version of the underlying direction field. Since $SO(2)$ is isomorphic to S^1, the suggested regularization scheme can be used for regularizing directions, such as *principal motion directions* in a video sequence. A reasonable choice for a data term that does not depend on the vector lengths would try to align the rotated first coordinate axis with the motion directions in the neighborhood,

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

where $\left(x_j, y_j, \left(v_j\right)_x, \left(v_j\right)_y\right)$ represent a sampled motion particle [35] in the video sequence (location and velocity), and $U_{i,j}$ represent elements of the solution u at each point.

In Figure 2 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 $\vec{v}(x,y) = \frac{(x,y)^T - \mathbf{c}}{\|(x,y)^T - \mathbf{c}\|}$, where $\mathbf{c}$ denotes the center of the image. 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 remains reasonable, due to the robust nature of total-variation regularization.

Another classical example of direction diffusion is in denoising of directions in fingerprint images. An example of direction diffusion on a fingerprint image taken from the Fingerprint Verification Competition datasets [1] can be seen in Figure 3. Adding a noise of $\sigma = 0.05$ to the image and estimating directions based on the structure tensor, we then smoothed the directions 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 30 with $\tilde{\lambda} = 10$, as well as the method suggested by Sagiv et al. [49] 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 and MSE of 0.0449. The results demonstrate the effectiveness of our method for direction diffusion, even in cases where the staircasing effect could have lead to unwanted artifacts.

4.2. $SE(n)$ **regularization.** An example of $SE(3)$ valued images can be obtained by doing local matches between two range scans obtained from a Kinect device. For each small surface patch from the depth image we use an iterative closest point algorithm [13, 6] to match the surface from the previous frame. This provides us with a field over $SE(3)$ over the image plane for every time frame. The method is described as Algorithm 2.

Algorithm 2 Regularized 3D rigid motion estimation

for each frame t, and the scanned 3D surface at time t, S_t **do**
  for each pixel $\mathbf{x}_i \in S_t$ **do**
    Crop a small patch around $\mathbf{x}_i$, $P_i = S_t \cap B_r(\mathbf{x}_i)$
    Estimate the rigid motion R_i, t_i that minimizes the L_2 error between P_i and S_{t-1}, by running the iterative closest point algorithm.
    Set $u_0(\mathbf{x}_i) = (\mathbf{R}_i, \mathbf{t}_i)$
  end for
  Perform regularization of u_0 using Algorithm 1.
end for

TABLE 1
Processing times (ms) for various sizes of images, with various iteration counts.

Outer iterations	15	15	25	50	100
GS iterations	1	3	1	1	1
320×240	49	63	81	160	321
640×480	196	250	319	648	1295
1920×1080	1745	2100	2960	5732	11560

We note that this measurement process is highly contaminated by non-Gaussian noise. Despite this highly level of noise we can use our algorithm to smooth this $SE(3)$ image, obtaining a scale-space of $SE(3)$ images, as shown in Figure 4. 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. Visualization is accomplished by projecting the embedded matrix onto 3 different representative vectors in $\mathbb{R}^{12}$.

In order to demonstrate the efficiency of this method and its parallelizable nature, we implement it using the CUDA framework, and measure the computational time required in order to obtain practical levels of convergence, at least 3 orders of magnitude. The computation times are shown in Table 1, for various image sizes and iterations. Most of the examples shown are with only 1 inner iteration since this has given us the fastest convergence, without artifacts in the final result. In the GPU implementation the polar decomposition was chosen for its simplicity and efficiency. In practice, one Gauss-Seidel iteration sufficed to update u. Using 15 outer iterations, practical convergence is achieved in 49 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 crucial. We note that we do not use here the information between more than two frames, as is often done in optical flow. Furthermore, using nonrigid deformation rather than ICP in order to compute u_0 leads to significantly better results using the same regularization scheme, as we demonstrated [44], including segmentation of articulated motion. Comparative discussion of the effect of various initial estimation methods is beyond the scope of this paper.

4.3. DT-MRI regularization. In Figure 5 we demonstrate a smoothing of DT-MRI data from [34], based on the scheme suggested in Section 3.6. We show an axial view of the brain, glyph-based visualization using Slicer3D [2], with anisotropy-based color coding. In this visualization, the color of the glyphs marks isotropic (red)

to anisotropic (blue/violet) tensors. The relative size of the ellipsoids denotes the amount of diffusion (trace of the tensor), and the directions of the ellipsoids' principal axes align with the eigenvectors of the tensor at each point.

The noise added is an additive Gaussian noise in each of the tensor elements with $\sigma = 0.1$. Note that while different noise models are often assumed for diffusion-weighted images, at high noise levels the Gaussian model is a reasonable approximation. Regularization with $\lambda = 30$ is able to restore a significant amount of the white matter structure. At such levels of noise, the TV-regularized data bias towards isotropic tensors (known as the *swell effect* [14]) is less significant. The RMS of the tensor representation was 0.0406 in the corrupted image and 0.0248 in the regularized image. An additional application of our method is to perform regularized reconstruction of DT-MRI signals from diffusion-weighted images (DWI). This is done by replacing the quadratic fidelity term with a fitting term based on the Stejskal-Tanner equation [51]. Demonstrating this application is shown in our technical report [46], as discussion of such reconstruction terms is beyond the scope of this paper.

5. Conclusions. We propose in this paper a general framework for matrix-valued image regularization. Our framework is based on the augmented-Lagrangian technique, and its reformulation in terms of split-Bregman iterations is shown. Using the augmented Lagrangian technique, we separate the optimization problem into a TV-regularization step and a projection step, both of which can be solved in an easy-to-implement and parallel way. Specifically, we show the efficiency and effectiveness of the resulting scheme through several examples whose data taken from $SO(2)$, $SE(3)$, and $SPD(3)$ respectively. Our algorithms allow real-time regularization for tasks in image analysis and computer vision.

In future work we intend to explore other applications for matrix-valued image regularization as well as generalize our method to other modalities and domains of the maps.

REFERENCES

[1] Fingerprints Verification Competition database.
[2] 3DSlicer software package.
[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 (2010), pp. 438–457.
[4] P. J. Basser, J. Mattiello, and D. LeBihan, *MR diffusion tensor spectroscopy and imaging*, Biophysical journal, 66 (1994), pp. 259–267.
[5] 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 (2009), pp. 297–308.
[6] P. J. Besl and N. D. McKay, *A method for registration of 3D shapes*, IEEE-TPAMI, 14 (1992), pp. 239–256.
[7] L. Bregman, *The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming*, USSR Computational Mathematics and Mathematical Physics, 7 (1967), pp. 200–217.
[8] X. Bresson and T. Chan, *Fast dual minimization of the vectorial total variation norm and applications to color image processing*, Inverse Problems and Imaging, 2 (2008), pp. 455–484.
[9] A. D. Bue, J. Xavier, L. Agapito, and M. Paladini, *Bilinear factorization via augmented Lagrange multipliers*, in ECCV, pp. 283–296, Springer-Verlag, 2010.
[10] 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 (1999), pp. 421–438.
[11] A. Chambolle, *An algorithm for total variation minimization and applications*, JMIV, 20 (2004), pp. 89–97.

[12] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, JMIV, 40 (2011), pp. 120–145.
[13] Y. Chen and G. Medioni, *Object modelling by registration of multiple range images*, Image Vision Comput., 10 (1992), pp. 145–155.
[14] R. Deriche, D. Tschumperle, and C. Lenglet, *DT-MRI estimation, regularization and fiber tractography*, in ISBI, pp. 9–12, 2004.
[15] R. Duits and B. Burgeth, *Scale spaces on Lie groups*, in SSVM, pp. 300–312, 2007.
[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., pp. 379–384, 2008.
[17] V. Duval, J.-F. Aujol, and L. A. Vese, *Mathematical modeling of textures: Application to color image decomposition with a projected gradient algorithm*, JMIV, 37 (2010), pp. 232–248.
[18] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 303–353.
[19] R. Fletcher, *Semi-definite matrix constraints in optimization*, SIAM J. on Cont. and Optimization, 23 (1985), pp. 493–513.
[20] O. Freifeld and M. J. Black, *Lie bodies: A manifold representation of 3D human shape*, in ECCV, pp. 1–14, 2012.
[21] W. Gibson, *On the least-squares orthogonalization of an oblique transformation*, Psychometrika, 27 (1962), pp. 193–195.
[22] B. Goldluecke, E. Strekalovskiy, and D. Cremers, *The natural vectorial variation which arises from geometric measure theory*, SIAM J. Imag. Sci., 5 (2012), pp. 537–563.
[23] T. Goldstein, X. Bresson, and S. Osher, *Geometric applications of the split Bregman method: Segmentation and surface reconstruction*, J. of Sci. Comp., 45 (1–3), pp. 272–293.
[24] T. Goldstein and S. Osher, *The split Bregman method for l1-regularized problems*, SIAM J. Imag. Sci., 2 (2009), pp. 323–343.
[25] Y. Gur and N. Sochen, *Fast invariant Riemannian DT-MRI regularization*, in ICCV, pp. 1–7, 2007.
[26] B. C. Hall, *Lie Groups, Lie Algebras,and Representations, An Elementary Introduction*, Springer, 2004.
[27] M. R. Hesteness, *Multipliers and gradient methods*, J. of Optimization Theory and Applications, 4 (1969), pp. 303–320.
[28] N. J. Higham, *Matrix nearness problems and applications*, in Applications of Matrix Theory, pp. 1–27, Oxford University Press, Oxford, 1989.
[29] R. Kimmel and N. Sochen, *Orientation diffusion or how to comb a porcupine*, special issue on PDEs in Image Processing, Comp. Vision, and Comp. Graphics, J. of Vis. Comm. and Image Representation, 13 (2002), pp. 238–248.
[30] N. Kiryati, T. Riklin-Raviv, Y. Ivanchenko, and S. Rochel, *Real-time abnormal motion detection in surveillance video*, in ICPR, pp. 1–4, 2008.
[31] R. Lai and S. Osher, *A splitting method for orthogonality constrained problems*, Technical Report CAM 12-39, UCLA, 2012.
[32] P. M. Larochelle, A. P. Murray, and J. Angeles, *On advances in robot kinematics*, chapter SVD and PD based projection metrics on SE(N), pp. 13–22, Kluwer, 2004.
[33] D. Lin, W. Grimson, and J. Fisher, *Learning visual flows: A Lie algebraic approach*, in CVPR, pp. 747–754, 2009.
[34] A. Lundervold, *On consciousness, resting state fMRI, and neurodynamics*, Nonlinear Biomed Phys, 4 Suppl 1, 2010.
[35] R. Mehran, B. E. Moore, and M. Shah, *A streakline representation of flow in crowded scenes*, in ECCV, 2010.
[36] M. Nicolescu and G. Medioni, *A voting-based computational framework for visual motion analysis and interpretation*, IEEE-TPAMI, 27 (2005), pp. 739–752.
[37] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, *An iterative regularization method for total variation-based image restoration*, Simul, 4 (2005), pp. 460–489.
[38] F. C. Park, J. E. Bobrow, and S. R. Ploen, *A Lie group formulation of robot dynamics*, Int. J. Rob. Res., 14 (1995), pp. 609–618.
[39] X. Pennec, P. Fillard, and N. Ayache, *A Riemannian framework for tensor computing*, IJCV, 66 (2006), pp. 41–66.
[40] P. Perona, *Orientation diffusions*, IEEE Trans. Image Process., 7 (1998), pp. 457–467.
[41] M. J. Powell, *Optimization*, chapter A method for nonlinear constraints in minimization problems, pp. 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 ECCV, pp. 577–590, Sep. 2010.
[44] G. Rosman, A. M. Bronstein, M. M. Bronstein, X.-C. Tai, and R. Kimmel, *Group-valued regularization for analysis of articulated motion*, in NORDIA workshop, ECCV, pp. 52–62, Berlin, Heidelberg, 2012. Springer-Verlag.
[45] 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.
[46] G. Rosman, Y. Wang, X.-C. Tai, R. Kimmel, and A. M. Bruckstein, *Fast regularization of matrix-valued images*, Technical Report CAM11-87, UCLA, 2011.
[47] G. Rosman, Y. Wang, X.-C. Tai, R. Kimmel, and A. M. Bruckstein, *Fast regularization of matrix-valued images*, in ECCV, pp. 173–186, Berlin, Heidelberg, 2012. Springer-Verlag.
[48] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D Letters, 60 (1992), pp. 259–268.
[49] N. A. Sochen, C. Sagiv, and R. Kimmel, *Stereographic combing a porcupine or studies on direction diffusion in image processing*, SIAM J. Appl. Math., 64 (2004), pp. 1477–1508.
[50] G. Steidl, S. Setzer, B. Popilka, and B. Burgeth, *Restoration of matrix fields by second-order cone programming*, Computing, 81 (2007), pp. 161–178.
[51] 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 (1965), pp. 288–292.
[52] X.-C. Tai and C. Wu, *Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model*, in SSVM, pp. 502–513, 2009.
[53] B. Tang, G. Sapiro, and V. Caselles, *Diffusion of general data on non-flat manifolds via harmonic maps theory: The direction diffusion case*, IJCV, 36 (2000), pp. 149–161.
[54] D. Tschumperle and R. Deriche, *Vector-valued image regularization with PDEs: A common framework for different applications*, IEEE-TPAMI, 27 (2005), pp. 506–517.
[55] P. Tseng, *Coordinate ascent for maximizing nondifferentiable concave functions*, LIDS-P 1940, MIT, 1988.
[56] O. Tuzel, F. Porikli, and P. Meer, *Learning on Lie-groups for invariant detection and tracking*, in CVPR, pp. 1–8, 2008.
[57] B. C. Vemuri, Y. Chen, M. Rao, T. McGraw, Z. Wang, and T. Mareci, *Fiber tract mapping from diffusion tensor MRI*, in VLSM, pp. 81–88. IEEE Computer Society, 2001.
[58] M. Žefran, V. Kumar, and C. Croke, *On the generation of smooth three-dimensional rigid body motions*, IEEE Transactions on Robotics and Automation, 14 (1998), pp. 576–589.
[59] 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.
[60] Z. Wen, D. Goldfarb, and W. Yin, *Alternating direction augmented Lagrangian methods for semidefinite programming*, CAAM TR09-42, Rice university, 2009.
[61] 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 (2010), pp. 300–339.
[62] W. Yin and S. Osher, *Error forgetting of Bregman iteration*, J. Sci. Comput., 54 (2013), pp. 684–695.

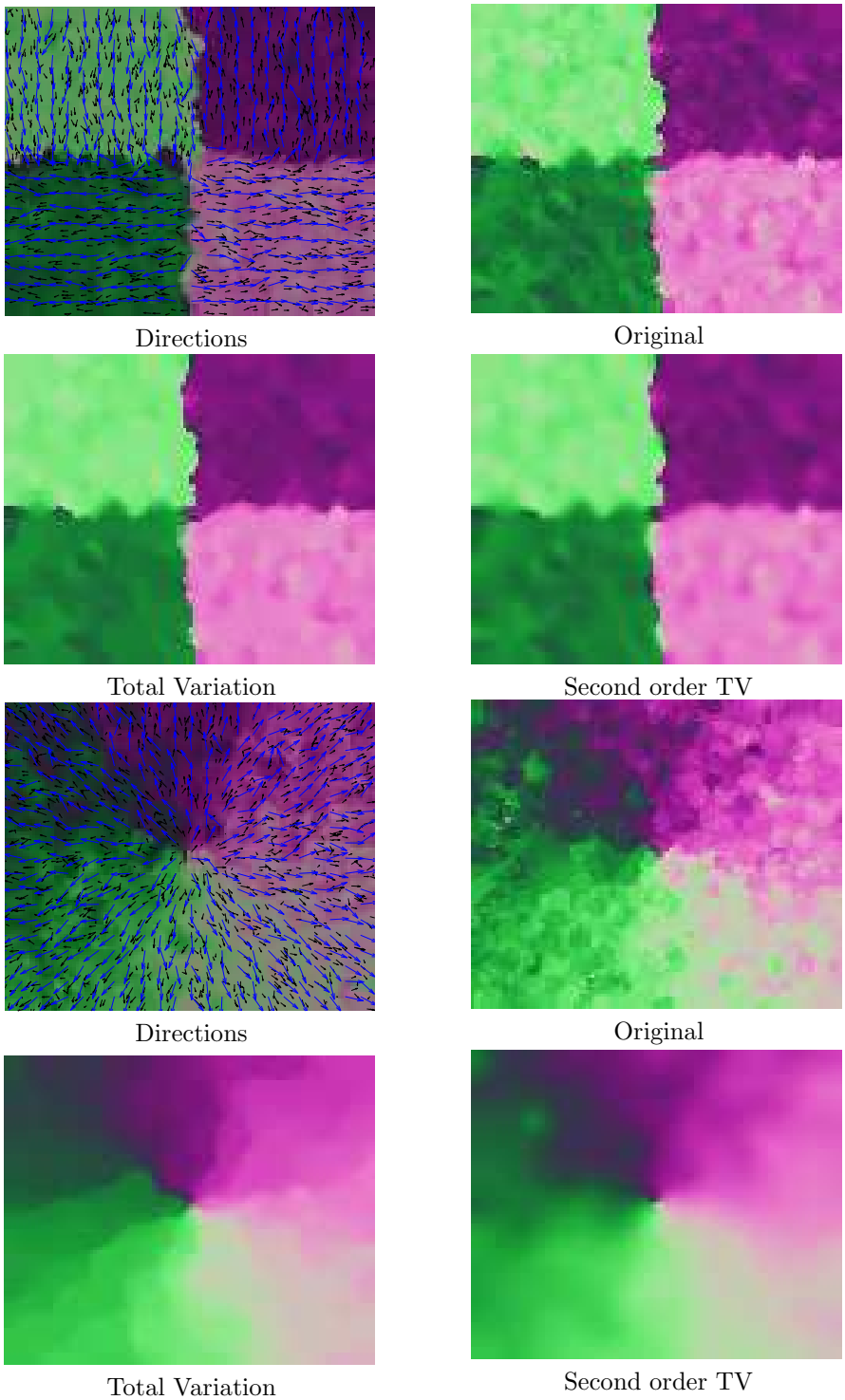

FIG. 2. *TV regularization of $SO(n)$ data. Left-to-right, top-to-bottom: a noisy, TV-denoised, and higher-order regularized (minimizing Equation 30) version of a piecewise constant $SO(2)$ image, followed by a expansion field direction image. Different colors mark different orientations of the initial/estimated dense field, black arrows signify the measured motion vectors, and blue arrows demonstrate the estimated field*

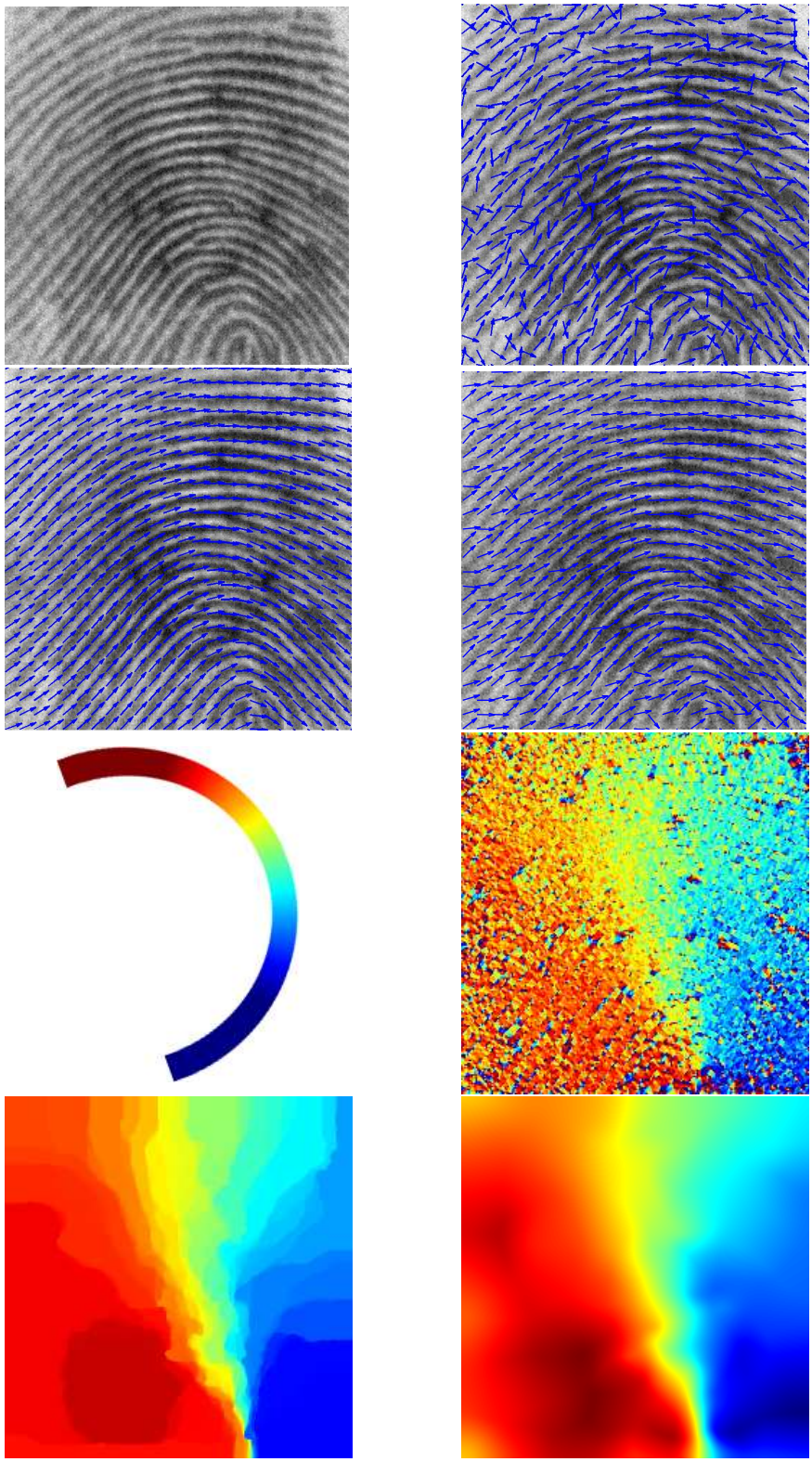

FIG. 3. *TV regularization of $SO(2)$ data based on fingerprint direction estimation. Top two rows,left-to-right: The fingerprint image with added Gaussian noise of $\sigma = 0.05$, the detected direction angles displayed as arrows, the detected directions after regularization with using a higher-order regularization term shown in Equation 30 with $\lambda = 6$, the regularization result by Sochen et al. [49]. Bottom two rows: color legend for the directionality images, initial estimated field, result of TV regularization with $\lambda = 3$, higher-order regularization with $\lambda = 6$.*

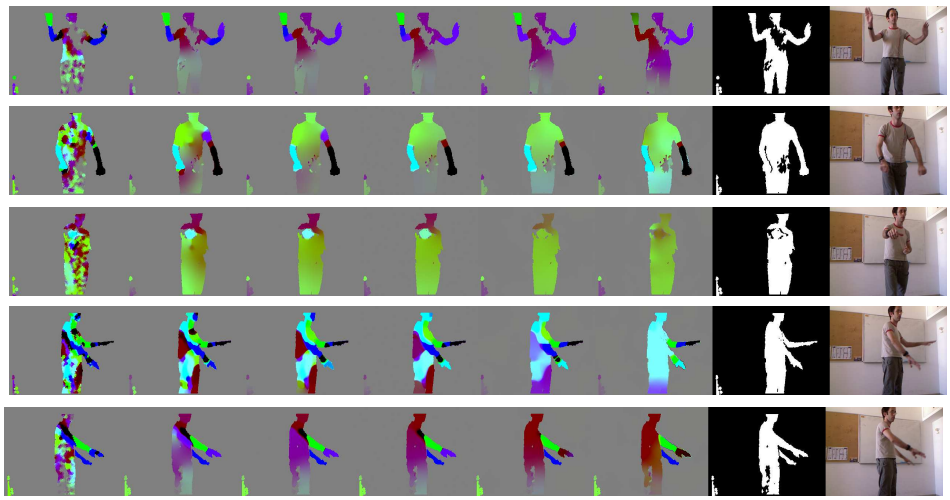

FIG. 4. *Regularization of $SE(3)$ images obtained from local ICP matching of the surface patch between consecutive Kinect depth frames. Left-to-right: diffusion scale-space obtained by different values of λ: $1.5, 1.2, 0.7, 0.2, 0.1, 0.05$, the foreground segmentation based on the depth, and an intensity image of the scene. Top-to-bottom: different frames from the depth motion sequence.*

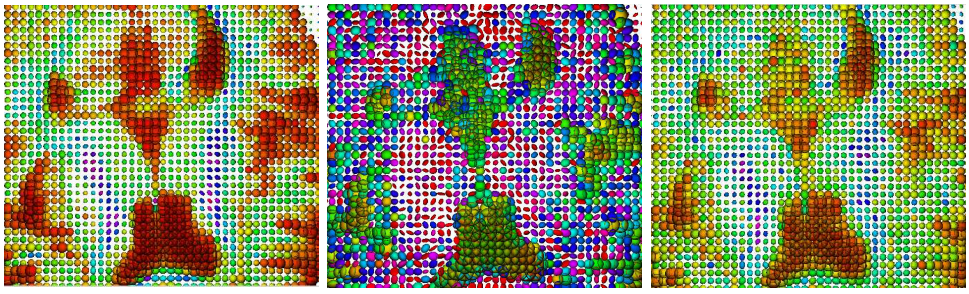

FIG. 5. *TV denoising of images with diffusion tensor data, visualized by 3D tensor ellipsoid glyphs colored by fractional anisotropy. Left-to-right: the original image, an image with added component-wise Gaussian noise of $\sigma = 0.1$, and the denoised image with $\lambda = 30$.*