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] Dept. of Computer Science
Technion - IIT
Haifa 32000 Israel
[2] Dept. of Mathematics
University of Bergen
Johaness Brunsgate 12
Bergen 5007 Norway

Abstract. Regularization of images with matrix-valued data is important in medical imaging, motion analysis and scene understanding. We propose a novel method for fast regularization of matrix group-valued images.
Using the augmented Lagrangian framework we separate total-variation regularization of matrix-valued images into a regularization and a projection steps. Both steps are computationally efficient and easily parallelizable, allowing real-time regularization of matrix valued images on a graphic processing unit.
We demonstrate the effectiveness of our method for smoothing several group-valued image types, with applications in directions diffusion, motion analysis from depth sensors, and DT-MRI denoising.

1 Introduction

Matrix Lie-group data, and specifically matrix-valued images have become an integral part of computer vision and image processing. Such representations have been found useful for tracking [35, 44], robotics, motion analysis, image processing and computer vision [10, 32, 34, 36, 47], as well as medical imaging [6, 31]. Specifically, developing efficient regularization schemes for matrix-valued images is of prime importance for image analysis and computer vision. This includes applications such as direction diffusion [25, 41, 46] and scene motion analysis [27] in computer vision, as well as diffusion tensor MRI (DT-MRI) regularization [7, 14, 21, 39, 42] in medical imaging.

In this paper we present an augmented Lagrangian method for efficient regularization of matrix-valued images with constraints on the singular values or eigenvalues of the matrices. Examples include the special-orthogonal, special-Euclidean, and symmetric positive-definite matrix groups. We show that the augmented Lagrangian technique allows us to separate the optimization process into a *total-variation* (TV, [37]) regularization, or higher-order regularization step, and an eigenvalues or singular values projection step, both of which 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 domains into one framework, where only the projection operator is slightly different according to the matrix group in question. While such an optimization problem could have been approached by general saddle-point solvers such as [12], the domain of our problem is not convex, requiring such algorithms to be revisited in order to prove their convergence.

We suggest using two sets of auxiliary fields with appropriate constraints. One field allows us to simplify the total-variation regularization operator as done, for example, in [11, 20, 40]. Another field separates the matrix manifold 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. In Section 4 we demonstrate a few results of our method, for regularization of $3D$ motion analysis, direction diffusion and diffusion tensor imaging. Section 5 concludes the paper.

2 A Short Introduction to Lie-Groups

Lie-groups are groups endowed with a differentiable manifold structure and an appropriate group action. Their structure allows us to define priors on Lie-group data in computer vision and has been the subject of intense research efforts, especially involving statistics of matrix-valued data [31], and regularization of group-valued images [42], as well as describing the dynamics of processes involving Lie-group data [27]. We briefly describe the Lie-groups our algorithm deals with, and refer the reader to the literature for an introduction to Lie-groups [22].

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 group $SPD(n)$ - This group is the group of symmetric positive definite matrices. This group has been studied extensively in control theory (see [17] for example), as well as in the context of diffusion tensor images [31], where the matrices are used to describe the diffusion coefficients along each direction. By definition, this group is given in matrix form as

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

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 optimization problem we consider is

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

where $\|\cdot\|$ is the Frobenius norm, u represents an element in an embedding of the Lie-group $\mathcal{G}$ into Euclidean space, specifically for the groups $SO(n), SE(n), SPD(n)$. We use the notation ∇u to denote the Jacobian of u, described as a column-stacked vector. The regularization term $\|u^{-1}\nabla u\|$ expresses smoothness in terms of the geometry of the Lie-group. 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 $\mathbb{R}^m, m = n(n+1)/2$.

For brevity's sake, we use the same notation to represent the Lie-group element, its matrix representation, and the embedding onto 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. In order to obtain a fast regularization scheme, we look instead at regularization of an embedding of the Lie-group elements into Euclidean space,

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

The rationale behind the different regularization term $\|\nabla u\|$ stems from the fact that $SO(n)$ and $SE(n)$ are isometries of Euclidean space, but such a regularization is possible whenever the data consists of nonsingular matrices, and has been used also for SPD matrices [45]. We refer the reader to our technical report [3] 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 [23, 33]. The suggested augmented Lagrangian optimization now reads

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

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 are trivial to implement in an efficient and parallel manner.

3.1 Minimization w.r.t. v

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

$$\begin{aligned}
&\underset{v\in\mathcal{G}}{\operatorname{argmin}} \frac{r}{2}\|v - u\|^2 + \text{tr}(\mu^T(u - v)) \\
&= \underset{v\in\mathcal{G}}{\operatorname{argmin}} \frac{r}{2}\left\|v - \left(\frac{\mu}{r} + u\right)\right\|^2 \qquad (7)\\
&= \underset{\mathcal{G}}{\operatorname{Proj}}\left(\frac{\mu}{r} + u\right),
\end{aligned}$$

where $\operatorname{Proj}_{\mathcal{G}}$ 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

$$\underset{u \in \mathbb{R}^m}{\operatorname{argmin}} \int \|\nabla u\| + \tilde{\lambda} \left\| u - \tilde{u}\left(u_0, v, \mu, r\right) \right\|^2 dx, \tag{8}$$

with $\tilde{u} = \frac{(2\lambda u_0 + rv + \mu)}{(2\lambda + r)}$. This problem can be solved via fast minimization techniques for TV regularization of vectorial images, such as [9, 16, 19]. We chose to use the augmented-Lagrangian TV algorithm [40], 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

$$\min_{\substack{u \in \mathbb{R}^m \\ p \in \mathbb{R}^{2m}}} \max_{\mu_2} \int \begin{bmatrix} \tilde{\lambda} \left\| u - \tilde{u}\left(u_0, v, \mu, r\right) \right\|^2 + \|p\| \\ + \mu_2^T (p - \nabla u) + \frac{r_2}{2} \|p - \nabla u\|^2 \end{bmatrix} dx. \tag{9}$$

We solve for u using the Euler-Lagrange equation,

$$2\tilde{\lambda}(u - \tilde{u}) + (\operatorname{div} \mu_2 + r_2 \operatorname{div} p) + \Delta u = 0, \tag{10}$$

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

$$\underset{p \in \mathbb{R}^{2m}}{\operatorname{argmin}} \int \|p\| + \mu_2^T p + \frac{r_2}{2} \|p - \nabla u\|^2, \tag{11}$$

with the closed-form solution [40]

$$p = \frac{1}{r_2} \max\left(1 - \frac{1}{\|w\|}, 0\right) w, w = r_2 \nabla u - \mu_2. \tag{12}$$

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

$$\begin{aligned} \mu^k &= \mu^{k-1} + r\left(v^k - u^k\right), \\ \mu_2^k &= \mu_2^{k-1} + r_2\left(p^k - \nabla u^k\right). \end{aligned} \tag{13}$$

An algorithmic description is summarized as Algorithm 1.

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 [18]. 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

$$\begin{aligned} v^{k+1} &= \operatorname*{Proj}_{SO(n)} \left(\frac{\mu}{r} + u^k\right) = \mathbf{U}(x)\mathbf{V}^T(x), \\ \mathbf{USV}^T &= \left(\frac{\mu}{r} + u^k\right). \end{aligned} \tag{14}$$

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 (10,12).
3:     Update $v^k(x)$, by projection onto the matrix group,
           – For $SO(n)$ matrices, according to Equation (14).
           – For $SE(n)$ matrices, according to Equation (15).
           – For $SPD(n)$ matrices, according to Equation (16).
4:     Update $\mu^k(x), \mu_2^k(x)$, according to Equation (13).
5: end for
```

Other possibilities include using the Euler-Rodrigues formula, quaternions, or the polar decomposition [26]. We note that the nonconvex domain $SO(n)$ prevents a global convergence proof. The algorithm, in the case of $\mathcal{G} = SO(n)$ and $\mathcal{G} = SE(n)$, can be made provably convergent using the method of Attouch et al. [5]. The details and proof are shown in our technical report [3].

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, with a similar partition for the Lagrange multipliers $\mu = (\mu_R, \mu_t)$. Updating v in step 3 of Algorithm 1 assumes the form

$$\begin{aligned} v_R^{k+1} &= \operatorname*{Proj}_{SO(n)}\left(\frac{\mu_R}{r} + u_R^k\right),\ v_t^{k+1} = \left(\frac{\mu_t}{r} + u_t^k\right) \\ v^{k+1} &= \operatorname*{Proj}_{SE(n)}(v^k) = (v_R^{k+1}, v_t^{k+1}). \end{aligned} \tag{15}$$

3.5 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 [24]. Many papers dealing with the 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

$$\begin{aligned} v^{k+1} &= \operatorname*{Proj}_{SPD(n)}(v^k) = \mathbf{U}(x)\operatorname{diag}\left(\hat{\boldsymbol{\lambda}}\right)\mathbf{U}^T(x), \\ \mathbf{U}\operatorname{diag}(\boldsymbol{\lambda})\mathbf{U}^T &= \left(\frac{\mu}{r} + u^k\right), \left(\hat{\boldsymbol{\lambda}}\right)_i = \max\left((\boldsymbol{\lambda})_i, 0\right), \end{aligned} \tag{16}$$

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 [43]. We refer the interested reader to our technical report [3]. An example of using the proposed method for DT-MRI denoising is shown in Section 4.

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. Several higher-order priors can be incorporated into our scheme that do not suffer from staircasing effects. One such possibile higher-order term generalizes the scheme presented by Wu and Tai [48], by replacing the per-element gradient operator with a Hessian operator. The resulting saddle-point problem becomes

$$\min_{\substack{u \in \mathbb{R}^m \\ p \in \mathbb{R}^{4m}, \\ v \in \mathcal{G}}} \max_{\mu_2} \int \begin{bmatrix} \|p\| + \tilde{\lambda} \left\| u - \tilde{u}\left(u_0, v, \mu, r\right) \right\|^2 \\ + \mu_2^T (p - Hu) + \frac{r_2}{2} \|p - Hu\|^2 \end{bmatrix} dx, \tag{17}$$

where H denotes the per-element Hessian operator. We show an example using the appropriately modified scheme in Figures 1,3

4 Numerical Results

As discussed above, the proposed algorithmic framework is considerable general and suitable for various applications. In this section, several examples from different applications are used to substantiate the effectiveness and efficiency of 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 [30, and references therein], vehicle control [15], crowd behaviour analysis [29], and other applications[32].

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 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 [29] in the video sequence, and $U_{i,j}$ represent elements of the solution u at each point.

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

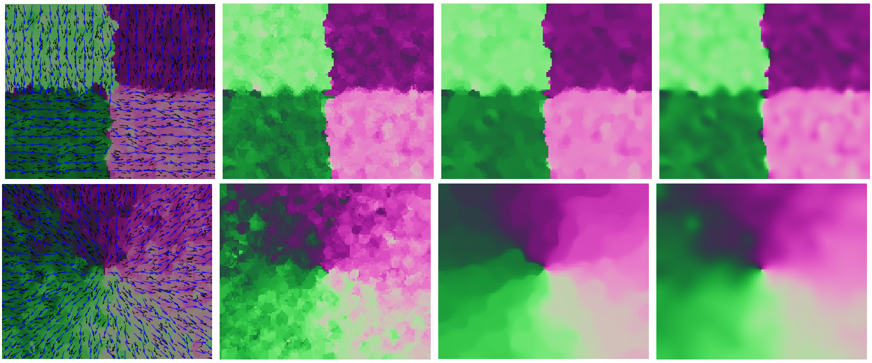

Fig. 1. TV regularization of $SO(n)$ data. Left-to-right, top-to-bottom: a noisy, TV-denoised, and higher-order regularized (minimizing Equation 17) version of a piecewise constant $SO(2)$ image, followed by a expansion field direction image. Different colors mark different orientations of the initial/estimated dense field, black arrows signify the measured motion vectors, and blue arrows demonstrate the estimated field

In Figure 2 we used the algorithm to obtain a smooth field of principal motion directions over a traffic sequence taken from the UCF crowd flow database [4]. Direction cues are obtained by initializing correlation-based trackers from arbitrary times and positions in the sequence, and observing all of them simultenaously. The result captures the main traffic lanes and shows the viability of our regularization for real data sequence.

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 3. Adding a noise of $\sigma = 0.05$ to the image and estimating directions based on the structure tensor, 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 17 with $\epsilon = 10$, as well as the method suggested by Sochen et al. [38] 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.

4.2 $SE(n)$ 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 algorithm[8] to match the surface from the previous frame. The

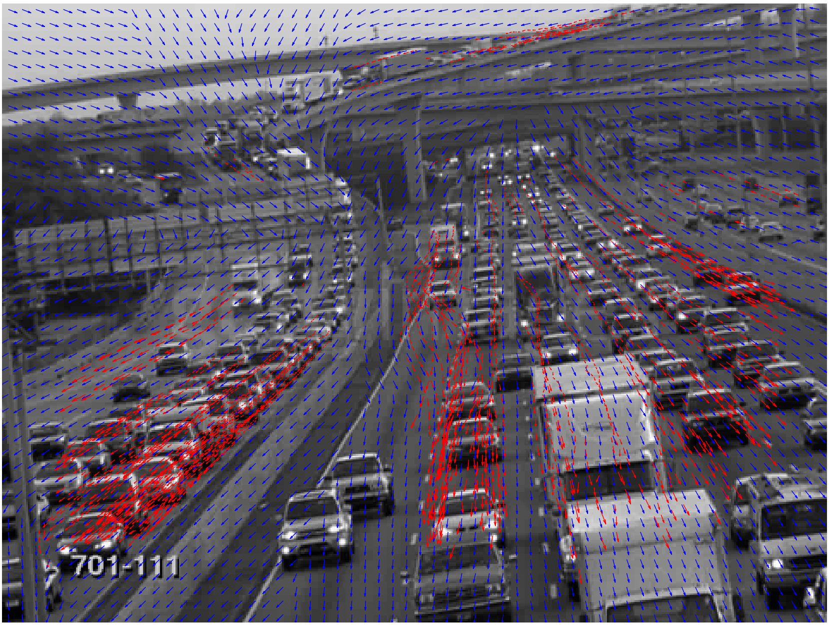

Fig. 2. Regularization of principal motion directions. The red arrows demonstrate measurements of motion cues based on a normalized cross-correlation tracker. Blue arrows demonstrate the regularized directions fields.

background is segmented by simple thresholding. The results from this tracking process over raw range footage are an inherently noisy measurements set. We use our algorithm to smooth this $SE(3)$ image, 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. 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, for various image sizes and iterations. 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.

4.3 DT-MRI regularization

In Figure 5 we demonstrate a smoothing of DT-MRI data from [28], based on the scheme suggested in Section 3.5. We show an axial view of the brain, glyph-based visualization using Slicer3D [2], with anisotropy-based color coding.

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

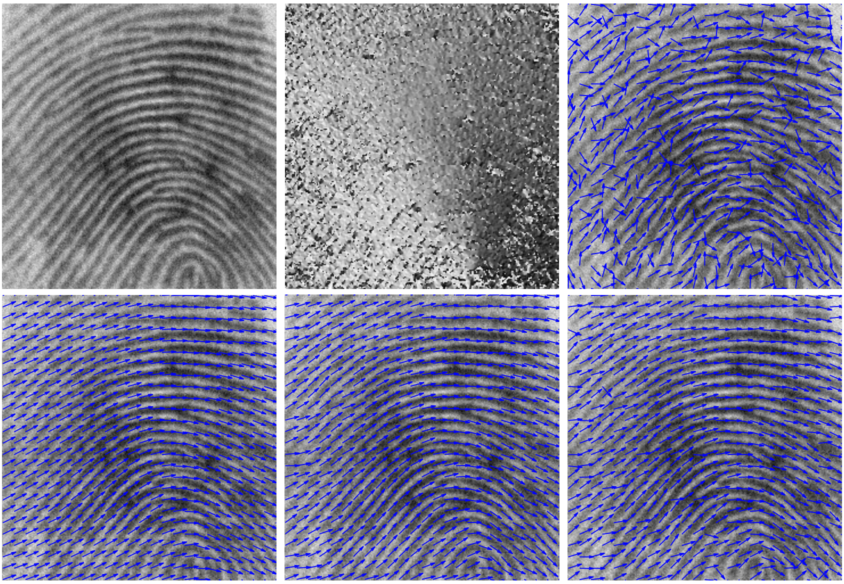

Fig. 3. TV regularization of $SO(2)$ data based on fingerprint direction estimation. Left-to-right, top-to-bottom: The fingerprint image with added Gaussian noise of $\sigma = 0.05$, the detected direction angles, the detected directions displayed as arrows, the detected directions after regularization with $\lambda = 3$, regularization results using a higher-order regularization term shown in Equation 17 with $\lambda = 6$, the regularization result by Sochen et al. [38].

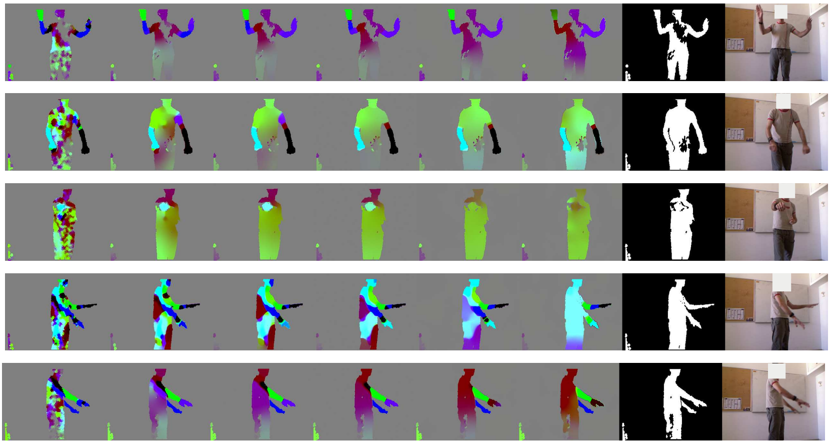

Fig. 4. Regularization of $SE(3)$ images obtained from local ICP matching of the surface patch between consecutive Kinect depth frames. Left-to-right: diffusion scale-space obtained by different values of λ: $1.5, 1.2, 0.7, 0.2, 0.1, 0.05$, the foreground segmentation based on the depth, and an intensity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

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

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

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* [13]) is less significant. The RMS of the tensor representation was 0.0406 in the corrupted image and 0.0248 in the regularized image. Similarly, regularized reconstruction of DT-MRI signals from diffusion-weighted images is also possible using our method, but is beyond the scope of this paper.

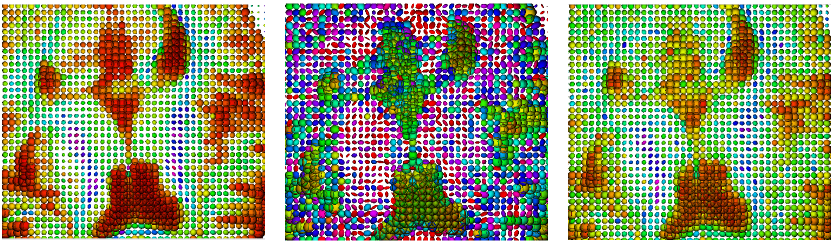

Fig. 5. TV denoising of images with diffusion tensor data, visualized by $3D$ tensor ellipsoid glyphs colored by fractional anisotropy. Left-to-right: the original image, an image with added component-wise Gaussian noise of $\sigma = 0.1$, and the denoised image with $\lambda = 30$.

5 Conclusions

In this paper, a general framework for regularization of matrix valued maps is proposed. Based on the augmented Lagrangian techniques, 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. To emphasize, for matrix-valued images, 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 types of maps, and data and noise models.

References

1. Fingerprints Verification Competition database.
2. 3DSlicer software package.

3. Fast regularization of matrix-valued images. Technical report, 2011. Anonymized version submitted as supplementary material.
4. S. Ali and M. Shah. A Lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *CVPR*, pages 1–6, 2007.
5. 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.
6. P. J. Basser, J. Mattiello, and D. LeBihan. MR diffusion tensor spectroscopy and imaging. *Biophysical journal*, 66(1):259–267, Jan. 1994.
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-TPAMI*, 14(2):239–256, 1992.
9. 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(4):455–484, 2008.
10. A. D. Bue, X. J. ao, L. Agapito, and M. Paladini. Bilinear factorization via augmented Lagrange multipliers. In *ECCV*, pages 283–296. Springer-Verlag, 2010.
11. A. Chambolle. An algorithm for total variation minimization and applications. *JMIV*, 20(1-2):89–97, 2004.
12. A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.
13. R. Deriche, D. Tschumperle, and C. Lenglet. DT-MRI estimation, regularization and fiber tractography. In *ISBI*, pages 9–12, 2004.
14. R. Duits and B. Burgeth. Scale spaces on Lie groups. In *SSVM*, pages 300–312, 2007.
15. 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.
16. 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(3):232–248, 2010.
17. R. Fletcher. Semi-definite matrix constraints in optimization. *SIAM J. on Cont. and Optimization*, 23(4):493–513, 1985.
18. W. Gibson. On the least-squares orthogonalization of an oblique transformation. *Psychometrika*, 27:193–195, 1962.
19. B. Goldluecke, E. Strekalovskiy, and D. Cremers. The natural vectorial variation which arises from geometric measure theory. *SIAM J. Imag. Sci.*, 2012.
20. 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):272–293.
21. Y. Gur and N. Sochen. Fast invariant Riemannian DT-MRI regularization. In *ICCV*, pages 1–7, 2007.
22. B. C. Hall. *Lie Groups, Lie Algebras,and Representations, An Elementary Introduction*. Springer, 2004.
23. M. R. Hesteness. Multipliers and gradient methods. *J. of Optimization Theory and Applications*, 4:303–320, 1969.
24. N. J. Higham. Matrix nearness problems and applications. In *Applications of Matrix Theory*, pages 1–27. Oxford University Press, Oxford, 1989.
25. 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*, 13:238–248, 2002.
26. 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.
27. D. Lin, W. Grimson, and J. Fisher. Learning visual flows: A Lie algebraic approach. In *CVPR*, pages 747–754, 2009.

28. A. Lundervold. On consciousness, resting state fMRI, and neurodynamics. *Nonlinear Biomed Phys*, 4 Suppl 1, 2010.
29. R. Mehran, B. E. Moore, and M. Shah. A streakline representation of flow in crowded scenes. In *ECCV*, 2010.
30. M. Nicolescu and G. Medioni. A voting-based computational framework for visual motion analysis and interpretation. 27:739–752, May 2005.
31. X. Pennec, P. Fillard, and N. Ayache. A Riemannian framework for tensor computing. *IJCV*, 66(1):41–66, 2006.
32. P. Perona. Orientation diffusions. *IEEE Trans. Image Process.*, 7(3):457–467, 1998.
33. M. J. Powell. *Optimization*, chapter A method for nonlinear constraints in minimization problems, pages 283–298. Academic Press, 1969.
34. I. U. Rahman, I. Drori, V. C. Stodden, D. L. Donoho, and P. Schroeder. Multiscale representations of manifold-valued data. Technical report, Stanford, 2005.
35. M. Raptis and S. Soatto. Tracklet descriptors for action modeling and video analysis. In *ECCV*, pages 577–590, Sep. 2010.
36. 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.
37. L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D Letters*, 60:259–268, 1992.
38. 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(5):1477–1508, 2004.
39. 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.
40. X.-C. Tai and C. Wu. Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model. In *SSVM*, pages 502–513, 2009.
41. B. Tang, G. Sapiro, and V. Caselles. Diffusion of general data on non-flat manifolds via harmonic maps theory: The direction diffusion case. *IJCV*, 36:149–161, 2000.
42. D. Tschumperle and R. Deriche. Vector-valued image regularization with PDEs: A common framework for different applications. *IEEE-TPAMI*, 27:506–517, 2005.
43. P. Tseng. Coordinate ascent for maximizing nondifferentiable concave functions. LIDS-P 1940, MIT, 1988.
44. O. Tuzel, F. Porikli, and P. Meer. Learning on Lie-groups for invariant detection and tracking. In *CVPR*, pages 1–8, 2008.
45. B. C. Vemuri, Y. Chen, M. Rao, T. McGraw, Z. Wang, and T. Mareci. Fiber tract mapping from diffusion tensor MRI. In *VLSM*, pages 81–88. IEEE Computer Society, 2001.
46. 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.
47. Z. Wen, D. Goldfarb, and W. Yin. Alternating direction augmented Lagrangian methods for semidefinite programming. CAAM TR09-42, Rice university, 2009.
48. 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.

Now we have reached the maximum size of the ECCV 2012 submission.