MADMM: A Generic Algorithm for Non-smooth Optimization on Manifolds

Artiom Kovnatsky(✉), Klaus Glashoff, and Michael M. Bronstein

Institute of Computational Science, Faculty of Informatics,
USI Universitá della Svizzera Italiana, Lugano, Switzerland
{artiom.kovnatsky,klaus.glashoff,michael.bronstein}@usi.ch

Abstract. Numerous problems in computer vision, pattern recognition, and machine learning are formulated as optimization with manifold constraints. In this paper, we propose the *Manifold Alternating Directions Method of Multipliers* (MADMM), an extension of the classical ADMM scheme for manifold-constrained non-smooth optimization problems. To our knowledge, MADMM is the first generic non-smooth manifold optimization method. We showcase our method on several challenging problems in dimensionality reduction, non-rigid correspondence, multi-modal clustering, and multidimensional scaling.

1 Introduction

A wide range of problems in machine learning, pattern recognition, computer vision, and signal processing is formulated as optimization problems where the variables are constrained to lie on some Riemannian manifold. For example, optimization on the *Grassman manifold* comes up in multi-view clustering [1] and matrix completion [2]. Optimization on the *Stiefel manifold* arises in eigenvalue-, assignment-, and Procrustes problems, and in 1-bit compressed sensing [3]. Problems involving *products of Stiefel manifolds* include coupled diagonalization with applications to shape correspondence [4] and manifold learning [5], and eigenvector synchronization with applications to sensor localization [6], structural biology [7] and structure from motion recovery [8]. Optimization on the *sphere* is used in principle geodesic analysis [9], a generalization of the classical PCA to non-Euclidean domains. Optimization over the manifold of *fixed-rank matrices* arises in maxcut problems [10], sparse principal component analysis [10], regression [11], matrix completion [12,13], and image classification [14]. *Oblique manifolds* are encountered in problems such as independent component analysis [15], blind source separation [16], and prediction of stock returns [17].

Though some instances of manifold optimization such as eigenvalues problems have been treated extensively in the distant past, the first general purpose algorithms appeared only in the 1990s [18]. With the emergence of numerous applications during the last decade, especially in the machine learning community, there has been an increased interest in general-purpose optimization on different manifolds [19], leading to several manifold optimization algorithms

B. Leibe et al. (Eds.): ECCV 2016, Part V, LNCS 9909, pp. 680–696, 2016.
DOI: 10.1007/978-3-319-46454-1_41

such as conjugate gradients [20], trust regions [21], and Newton [18,22]. Boumal *et al.* [23] released the MATLAB package Manopt, as of today the most complete generic toolbox for smooth optimization on various manifolds.

In this paper, we are interested in manifold-constrained minimization of *non-smooth* functions, such as nuclear, L_1, or $L_{2,1}$ matrix norms. Recent examples of such problems include robust PCA [24], compressed eigenmodes [25,26], robust multidimensional scaling [27], synchronization of rotation matrices [28], and functional correspondence [29,30].

Prior Work. Broadly speaking, optimization methods for non-smooth functions break into three classes of approaches. First, *smoothing* methods replace the non-differentiable objective function with its smooth approximation [31]. Such methods typically suffer from a tradeoff between accuracy (how far is the smooth approximation from the original objective) and convergence speed (less smooth functions are usually harder to optimize). A second class of methods use *subgradients* as a generalization of derivatives of non-differentiable functions. In the context of manifold optimization, several subgradient approaches have been proposed [16,32–34]. The third class of methods are *splitting* approaches, studied mostly for problems involving the minimization of matrix functions with orthogonality constraints. Lai and Osher proposed the method of splitting orthogonal constraints (SOC) based on the Bregman iteration [35]. A similar method was independently developed in [36]. Neumann *et al.* [26] used a different splitting scheme for the same class of problems.

Contributions. In this paper, we propose *Manifold Alternating Direction Method of Multipliers* (MADMM), an extension of the classical ADMM scheme [37] for manifold-constrained non-smooth optimization problems. The core idea is a splitting into a smooth problem with manifold constraints and a non-smooth unconstrained optimization problem. We stress that while very simple, to the best of our knowledge we are the first to employ such a splitting, which leads to a general optimization method. Our method has a number of advantages common to ADMM approaches. First, it is very simple to grasp and implement. Second, it is generic and not limited to a specific manifold, as opposed to e.g. [26,35] developed for the Stiefel manifold, or [16] developed for the oblique manifold. Third, it makes very few assumptions about the properties of the objective function. Fourth, in some settings, our method lends itself to parallelization on distributed computational architectures [38]. Finally, our method demonstrates faster convergence than previous methods in a broad range of applications.

2 Manifold Optimization

The term *manifold-* or *manifold-constrained optimization* refers to a class of problems of the form

$$\min_{X \in \mathcal{M}} f(X), \tag{1}$$

where f is a smooth real-valued function, X is an $m \times n$ real matrix, and $\mathcal{M}$ is some Riemannian submanifold of $\mathbb{R}^{m \times n}$. The manifold is not a vector space and

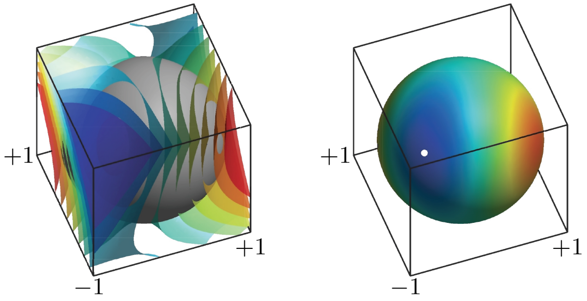


Fig. 1. The minimum eigenvalue problem $\min_{x \in \mathbb{R}^n} x^\top A x$ s.t. $x^\top x = 1$ is a simple example of a manifold optimization problem. Left: level sets of the cost function $x^\top A x$ for a random symmetric 3×3 matrix A. The manifold constraint (unit sphere $\{x \in \mathbb{R}^3 : x^\top x = 1\}$) is shown in grey. Right: values of the cost function on the manifold of feasible solutions. A minimizer (white dot) corresponds to the smallest eigenvector of A. Note that there are two minimizers in this example due to the sign ambiguity of the eigenvectors (the other minimizer is on the back of the sphere).

has no global system of coordinates, however, locally at point X, the manifold is homeomorphic to a Euclidean space referred to as the *tangent space* $T_X\mathcal{M}$.

The main idea of manifold optimization is to treat the objective as a function $f : \mathcal{M} \rightarrow \mathbb{R}$ defined on the manifold, and perform descent on the manifold itself rather than in the ambient Euclidean space (see a toy example in Fig. 1). On a manifold, the *intrinsic* (Riemannian) gradient $\nabla_\mathcal{M} f(X)$ of f at point X is a vector in the tangent space $T_X\mathcal{M}$ that can be obtained by projecting the standard (Euclidean) gradient $\nabla f(X)$ onto $T_X\mathcal{M}$ by means of a *projection* operator P_X (see an illustration below). A step along the intrinsic gradient direction is performed in the tangent plane. In order to obtain the next iterate, the point in the tangent plane is mapped back to the manifold by means of a *retraction* operator R_X, which is typically an approximation of the *exponential map*. For many manifolds, the projection P and retraction R operators have a closed form expression.

A conceptual gradient descent-like manifold optimization is presented in Algorithm 1. For a comprehensive introduction to manifold optimization, the reader is referred to [19].

3 Manifold ADMM

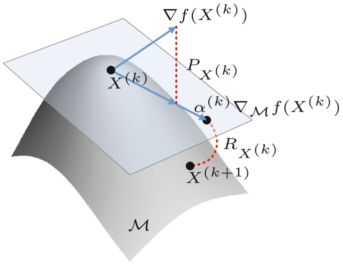
Let us now consider general problems of the form

$$\min_{X \in \mathcal{M}} f(X) + g(AX), \tag{2}$$

where f and g are smooth and non-smooth real-valued functions, respectively, A is a $k \times m$ matrix, and the rest of the notation is as in problem (1). Examples of g often used in machine learning applications are nuclear-, L_1-, or $L_{2,1}$-norms.

```
repeat
    Compute the extrinsic gradient $\nabla f(X^{(k)})$
    Projection: $\nabla_{\mathcal{M}} f(X^{(k)}) = P_{X^{(k)}}(\nabla f(X^{(k)}))$
    Compute the step size $\alpha^{(k)}$ along the descent direction
    Retraction:
    $X^{(k+1)} = R_{X^{(k)}}(-\alpha^{(k)} \nabla_{\mathcal{M}} f(X^{(k)}))$
until convergence;
```

Algorithm 1. Conceptual algorithm for smooth optimization on manifold $\mathcal{M}$.



Because of non-smoothness of the objective function, Algorithm 1 cannot be used directly to minimize (2).

In this paper, we propose treating this class of problems using the Alternating Directions Method of Multipliers (ADMM). The key idea is that problem (2) can be equivalently formulated as

$$\min_{X \in \mathcal{M}, Z \in \mathbb{R}^{k \times n}} f(X) + g(Z) \quad \text{s.t.} \quad Z = AX \tag{3}$$

by introducing an artificial variable Z and a linear constraint. The method of multipliers [39,40], applied to only the linear constraints in (3), leads to the minimization problem

$$\min_{X \in \mathcal{M}, Z \in \mathbb{R}^{k \times n}} f(X) + g(Z) + \tfrac{\rho}{2} \|AX - Z + U\|_{\mathrm{F}}^2 \tag{4}$$

where $\rho > 0$ and $U \in \mathbb{R}^{k \times n}$ have to be chosen and updated appropriately (see below). This formulation now allows splitting the problem into two optimization sub-problems w.r.t. to X and Z, which are solved in an alternating manner, followed by an updating of U and, if necessary, of ρ. Observe that in the first sub-problem w.r.t. X we minimize a *smooth* function with manifold constraints, and in the second sub-problem w.r.t. Z we minimize a non-smooth function without manifold constraints. Thus, the problem breaks down into two well-known sub-problems. This method, which we call *Manifold Alternating Direction Method of Multipliers* (MADMM), is summarized in Algorithm 2.

Note that MADMM is extremely simple and easy to implement. The X-step is the setting of Algorithm 1 and can be carried out using any standard smooth manifold optimization method. Similarly to common implementation of ADMM algorithms, there is no need to solve the X-step problem *exactly*; instead, only a few iterations of manifold optimization are done. Furthermore, for some manifolds and some functions f, the X-step has a closed-form solution. The implementation of the Z-step depends on the non-smooth function g, and in many cases has a closed-form expression: for example, when g is the L_1-norm, the Z-step boils down to simple shrinkage, and when g is nuclear norm, the Z-step is performed by singular value shrinkage[1]. ρ is the only parameter of

[1] More generally, it is a proximity operator of $\frac{1}{\rho} g(Z)$ at $AX + U$.

```
Initialize $k \leftarrow 1$, $Z^{(1)} = AX^{(1)}$, $U^{(1)} = 0$.
repeat
    X-step: $X^{(k+1)} = \operatorname*{argmin}_{X \in \mathcal{M}} f(X) + \frac{\rho}{2}\|AX - Z^{(k)} + U^{(k)}\|_\mathrm{F}^2$
    Z-step: $Z^{(k+1)} = \operatorname*{argmin}_{Z} g(Z) + \frac{\rho}{2}\|AX^{(k+1)} - Z + U^{(k)}\|_\mathrm{F}^2$
    $U^{(k+1)} = U^{(k)} + AX^{(k+1)} - Z^{(k+1)}$
    $k \leftarrow k + 1$
until convergence;
```

Algorithm 2. Generic MADMM method for non-smooth optimization on manifold $\mathcal{M}$.

the algorithm and its choice is not critical for convergence. In our experiments, we used a rather arbitrary fixed value of ρ, though in the ADMM literature it is common to adapt ρ at each iteration, e.g. using the strategy described in [38].

Convergence. Our MADMM belongs to the class of multiplier algorithms that can be considered as 'methods with partial elimination of constraints' [41] and as 'augmented Lagrangian methods with general lower-level constraints' [42]. We note that the convergence results of [41,42] do not apply in our case due to non-differentiability of the function g in (2). Furthermore, MADMM is an alternating method and thus is not covered by theoretical results on 'pure' multiplier methods. An avenue for obtaining convergence results for (a regularized version of) MADMM is the recently developed theory by [43], which is applicable to non convex and non-differentiable functions f and g. Attouch et al. [43] show convergence results for the class of *semi algebraic objects*, which includes Stiefel and other matrix manifolds. Wang et al. prove global convergence of ADMM in convex and non-smooth scenarios, however the non-smooth and non-convex parts should belong to a specific class of functions (piecewise linear functions, ℓ_q quasi-norms ($0 \leq q \leq 1$), etc.) [44], which limits the use of their convergence results. We defer a deeper study of convergence properties to future work.

4 Results and Applications

In this section, we show experimental results providing a numerical evaluation of our approach on several challenging applications from the domains of dimensionality reduction, pattern recognition, and manifold learning. All our experiments were implemented in MATLAB; we used the conjugate gradients and trust regions solvers from the Manopt toolbox [23] for the X-step. Time measurements were carried out on a PC with Intel Xeon 2.4 GHz CPU.

4.1 Compressed Modes

Problem Setting. Our first application is the computation of compressed modes, an approach for constructing localized Fourier-like bases [25]. Let us be given a manifold $\mathcal{S}$ with a Laplacian Δ, where in this context, 'manifold' can refer to both continuous or discretized manifolds of any dimension, represented as graphs, triangular meshes, etc., and should not be confused with the matrix manifolds we have discussed so far referring to manifold-constrained optimization problems. Here, we assume that the manifold is sampled at n points and the Laplacian is represented as an $n \times n$ sparse symmetric matrix. In many machine learning applications such as spectral clustering [45], non-linear dimensionality reduction, and manifold learning [46], one is interested in finding the first k eigenvectors of the Laplacian $\Delta\Phi = \Phi\Lambda$, where $\Phi = (\phi_1, \ldots, \phi_k)$ is the $n \times k$ matrix of the first eigenvectors arranged as columns, and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_k)$ is the diagonal $k \times k$ matrix of the corresponding eigenvalues.

The first k eigenvectors of the Laplacian can be computed by minimizing the *Dirichlet energy* with orthonormality constraints

$$\min_{\Phi \in \mathbb{R}^{n \times k}} \ \mathrm{tr}(\Phi^\top \Delta \Phi) \quad \text{s.t.} \quad \Phi^\top \Phi = I. \tag{5}$$

Laplacian eigenfunctions form an orthonormal basis on the Hilbert space $L^2(\mathcal{S})$ with the standard inner product, and are a generalization of the Fourier basis to non-Euclidean domains. The main disadvantage of such bases is that its elements are globally supported. Ozoliņš *et al.* [25] proposed a construction of localized quasi-eigenbases by solving

$$\min_{\Phi \in \mathbb{R}^{n \times k}} \ \mathrm{tr}(\Phi^\top \Delta \Phi) + \mu \|\Phi\|_1 \quad \text{s.t.} \quad \Phi^\top \Phi = I, \tag{6}$$

where $\mu > 0$ is a parameter. The L_1-norm (inducing sparsity of the resulting basis) together with the Dirichlet energy (imposing smoothness of the basis functions) lead to orthogonal basis functions, referred to as *compressed modes* that are localized and approximately diagonalize Δ.

Lai and Osher [35] and Neumann *et al.* [26] proposed two different splitting methods for solving problem (6). Lai *et al.* [35] solves (6) by the *splitting orthogonality constraint (SOC)*, introducing two additional variables $Q = \Phi$ and $P = \Phi$ so that (6) is equivalent to the following constrained optimization problem,

$$\min_{\Phi, \mathrm{Q}, \mathrm{P} \in \mathbb{R}^{n \times k}} \ \mathrm{tr}(\Phi^\top \Delta \Phi) + \mu \|\mathrm{Q}\|_1 \quad \text{s.t.} \quad Q = \Phi, \ \ P = \Phi, \ \ P^\top P = I, \tag{7}$$

solved by alternating minimization on Φ, P, and Q (Algorithm 3).

Solution. Here, we realize that problem (6) is an instance of manifold optimization on the Stiefel manifold $\mathbb{S}(n,k) = \{X \in \mathbb{R}^{n \times k} : X^\top X = I\}$ and solve it using MADMM, which assumes in this setting the form of Algorithm 4. The X-step involves optimization of a smooth function on the Stiefel manifold and can be carried out using standard manifold optimization algorithms; we use conjugate gradients and trust regions solvers. The Z-step requires the minimization of the sum of L_1- and L_2-norms, a standard problem in signal processing that has an explicit solution by means of thresholding (using the shrinking operator). In all our experiments, we used the parameter $\rho = 2$ for MADMM. For comparison

```
Input $n \times n$ Laplacian matrix $\varDelta$, parameter $\mu > 0$
Output $n \times k$ matrix $\varPhi$ of the first compressed modes of $\varDelta$
Initialize $k \leftarrow 1$, $\varPhi^{(1)}$, $P^{(1)} = Q^{(1)} = \varPhi^{(1)}$, $U^{(1)} = V^{(1)} = 0$
repeat
    $\varPhi^{(k+1)} = \underset{\varPhi}{\operatorname{argmin}} \operatorname{tr}(\varPhi^\top \varDelta \varPhi) + \frac{\rho}{2}\|\varPhi - Q^{(k)} + U^{(k)}\|_{\mathrm{F}}^2 + \frac{\rho'}{2}\|\varPhi - P^{(k)} + V^{(k)}\|_{\mathrm{F}}^2$
    $Q^{(k+1)} = \underset{Q}{\operatorname{argmin}}\, \mu\|Q\|_1 + \frac{\rho}{2}\|\varPhi^{(k+1)} - Q + U^{(k)}\|_{\mathrm{F}}^2$
    $P^{(k+1)} = \underset{P:P^\top P = I}{\operatorname{argmin}}\ \frac{\rho'}{2}\|\varPhi^{(k+1)} - P + V^{(k)}\|_{\mathrm{F}}^2$
    $U^{(k+1)} = U^{(k)} + \varPhi^{(k+1)} - Q^{(k+1)}$
    $V^{(k+1)} = V^{(k)} + \varPhi^{(k+1)} - P^{(k+1)}$
    $k \leftarrow k + 1$
until convergence;
```

Algorithm 3. SOC method [35] for computing compressed modes.

```
Input $n \times n$ Laplacian matrix $\varDelta$, parameter $\mu > 0$
Output $n \times k$ matrix $\varPhi$ of the first compressed modes of $\varDelta$
Initialize $k \leftarrow 1$, $\varPhi^{(1)} \leftarrow$random orthonormal matrix, $Z^{(1)} = \varPhi^{(1)}$, $U^{(1)} = 0$
repeat
    $\varPhi^{(k+1)} = \underset{\varPhi \in \mathbb{S}(n,k)}{\operatorname{argmin}} \operatorname{tr}(\varPhi^\top \varDelta \varPhi) + \frac{\rho}{2}\|\varPhi - Z^{(k)} + U^{(k)}\|_{\mathrm{F}}^2$
    $Z^{(k+1)} = \operatorname{Shrink}_{\frac{\mu}{\rho}}(\varPhi^{(k+1)} + U^{(k)})$
    $U^{(k+1)} = U^{(k)} + \varPhi^{(k+1)} - Z^{(k+1)}$
    $k \leftarrow k + 1$
until convergence;
```

Algorithm 4. MADMM method for computing compressed modes. $\operatorname{Shrink}_\alpha(x) = \frac{x}{\|x\|}\max\{0, \|x\| - \alpha\}$ is the shrinkage operator.

with the method of [35], we used the code provided by the authors, and implemented the method of [26] ourselves. All the methods were initialized by the same random orthonormal $n \times k$ matrix $\varPhi$.

Results. To study the behavior of ADMM, we used a simple 1D problem with a Euclidean Laplacian constructed on a line graph with n vertices. Figure 3 (top left) shows the convergence of MADMM with different random initializations. Figure 3 (top right) shows the convergence of MADMM using different solvers and number of iterations in the X-step. We did not observe any significant change in the behavior. Figure 3 (bottom left) studies the scalability of different algorithms, speaking clearly in favor of MADMM compared to the methods of [26,35]. Figure 3 (bottom right) shows the convergence of different methods for the computation of compressed modes on a triangular mesh of a human sampled at 8 K vertices (see examples in Fig. 2). MADMM shows the best performance among the compared methods.

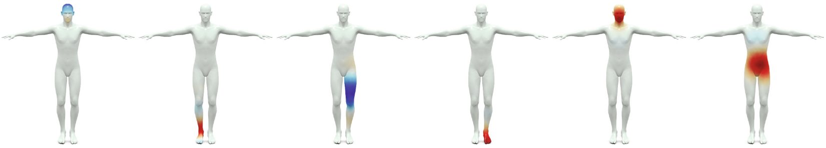


Fig. 2. First six compressed modes computed on a human mesh containing $n = 8\,\mathrm{K}$ points computed using MADMM. Parameter $\mu = 10^{-3}$ and three manifold optimization iterations in the X-step were used in these experiments.

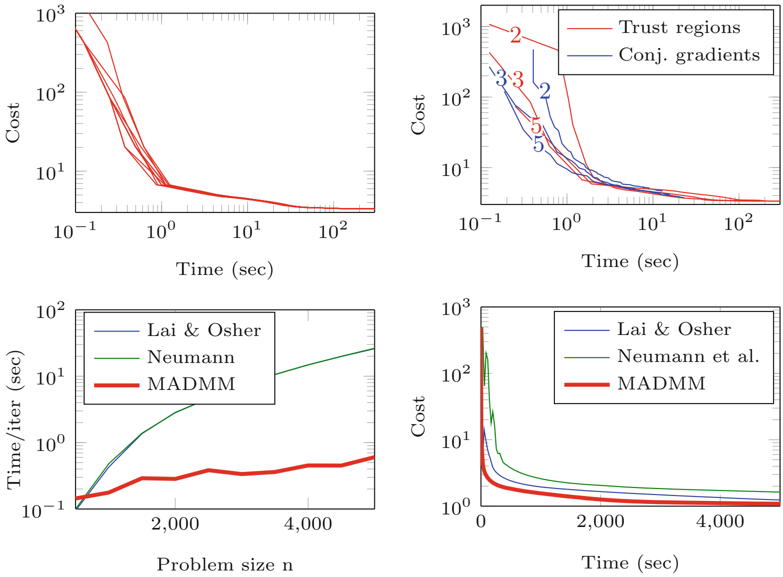


Fig. 3. Compressed modes problem. Top left: convergence of MADMM on a problem of size $n = 500, k = 10$ with different random initialization. Top right: convergence of MADMM using different solvers and number of iterations at X-step on the same problem. Bottom left: scalability of different methods; shown is time/iteration on a problem of different size (fixed $k = 10$ and varying n). Bottom right: comparison of convergence of different splitting methods and MADMM on a problem of size $n = 8\,\mathrm{K}$.

4.2 Functional Correspondence

Problem Setting. Our second problem is coupled diagonalization, which is used for finding functional correspondence between manifolds [47] and multi-view clustering [5]. Let us consider a collection of L manifolds $\{\mathcal{S}_i\}_{i=1}^{L}$, each discretized at n_i points and equipped with a Laplacian Δ_i represented as an $n_i \times n_i$ matrix. The *functional correspondence* between manifolds $\mathcal{S}_i$ and $\mathcal{S}_j$ is an $n_j \times n_i$ matrix T_{ij} mapping functions from $L^2(\mathcal{S}_i)$ to $L^2(\mathcal{S}_j)$. It can be efficiently approximated using the first k Laplacian eigenvectors as $T_{ij} \approx \Phi_j X_{ij} \Phi_i^\top$, where X_{ij} is the $k \times k$ matrix translating Fourier coefficients from basis Φ_i to basis Φ_j, represented as $n_i \times k$ and $n_j \times k$ matrices, respectively. Imposing a further assumption that T_{ij} is volume-preserving, X_{ij} must be an orthonormal matrix [47], which will be approximated by the product of two orthogonal matrices. For each pair of manifolds $\mathcal{S}_i, \mathcal{S}_j$, we assume to be given a set of q_{ij} functions in $L^2(\mathcal{S}_i)$ arranged as columns of an $n_i \times q_{ij}$ matrix F_{ij} and the corresponding functions in $L^2(\mathcal{S}_j)$ represented by the $n_j \times q_{ij}$ matrix G_{ij}. The correspondence between all the manifolds can be established by solving the problem

$$\min_{X_1,\ldots,X_L} \sum_{i \neq j} \|F_{ij}^\top \Phi_i X_i - G_{ij}^\top \Phi_j X_j\|_{2,1} + \mu \sum_{i=1}^{L} \mathrm{tr}(X_i^\top \Lambda_i X_i) \quad \text{s.t. } X_i^\top X_i = I. \tag{8}$$

The $L_{2,1}$-norm $\|A\|_{2,1} = \sum_j \left(\sum_i a_{ij}^2\right)^{1/2}$ allows to cope with outliers in the correspondence data [28,29]. The problem can be interpreted as simultaneous diagonalization of the Laplacians $\Delta_1, \ldots, \Delta_L$ [5]. As correspondence data F, G, one can use point-wise correspondence between some known 'seeds', or, in computer graphics applications, some shape descriptors [47]. Geometrically, the matrices X_i can be interpreted as rotations of the respective bases, and the problem tries to achieve a coupling between the bases $\hat{\Phi}_i = \Phi_i X_i$ while making sure that they approximately diagonalize the respective Laplacians.

Solution. Here, we consider problem (8) as optimization on a product of L Stiefel manifolds, $(X_1, \ldots, X_L) \in \mathbb{S}^L(k,k)$ and solve it using the MADMM method. The X-step of MADMM was performed using four iterations of the manifold conjugate gradients solver. As in the previous problem, the Z-step boils down to simple shrinkage. We used $\rho = 1$ and initialized all $X_i = I$.

```
Input $n_i \times q_{ij}$, $n_j \times q_{ij}$ corresponding matrices $F_{ij}, G_{ij}$, respectively;
$n_i \times k$ eigenbases matrices $\Phi_i$, $k \times k$ diagonal matrices $\Lambda_i$ of corresponding
eigenvalues, and parameter $\mu > 0$
Output orthonormal matrices $X_1, \ldots, X_L$ aligning the bases $\Phi_1, \ldots, \Phi_L$
and allowing to express the functional correspondences as
$T_{ij} \approx \Phi_j X_i X_j^\top \Phi_i^\top$
Initialize $k \leftarrow 1$, $X_i^{(1)} \leftarrow I$, $Z_{ij}^{(1)} \leftarrow F_{ij}^\top \Phi_i X_i^{(1)} - G_{ij}^\top \Phi_j X_j^{(1)}$, $U_{ij}^{(1)} \leftarrow I$
repeat
    $(X_1^{(k+1)}, \ldots, X_L^{(k+1)}) =$
    $\operatorname*{argmin}_{X_i \in \mathbb{S}(k,k)} \sum_i \mathrm{tr}(X_i^\top \Lambda_i X_i) + \frac{\rho}{2} \sum_{i \neq j} \|F_{ij}^\top \Phi_i X_i - G_{ij}^\top \Phi_j X_j - Z_{ij}^{(k)} + U_{ij}^{(k)}\|_F^2$
    $Z_{ij}^{(k+1)}(:,l) =$
    $\mathrm{Shrink}_{\frac{1}{\mu\rho}}(F_{ij}^\top \Phi_i X_i^{(k+1)}(:,l) - G_{ij}^\top \Phi_j X_j^{(k+1)}(:,l) + U_{ij}^{(k)}(:,l))$
    $U_{ij}^{(k+1)} = U_{ij}^{(k)} + F_{ij}^\top \Phi_i X_i^{(k+1)} - G_{ij}^\top \Phi_j X_j^{(k+1)} - Z_{ij}^{(k+1)}$
    $k \leftarrow k + 1$
until convergence;
```

Algorithm 5. MADMM method for functional correspondence problem. $X(:,l)$ denotes the lth column of matrix X.

Results. We computed functional correspondences between $L = 6$ human 3D shapes from the TOSCA dataset [48] using $k = 25$ basis functions and $q = 25$

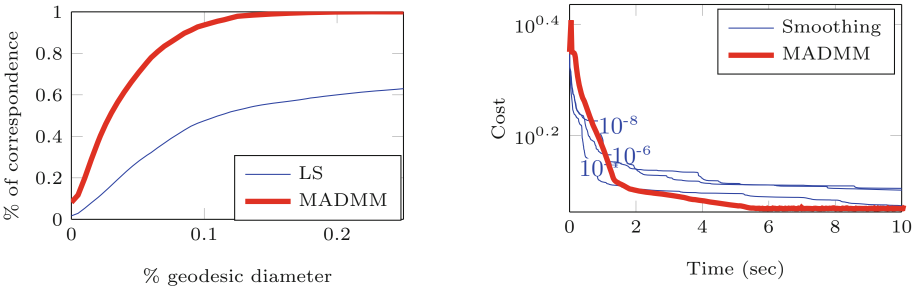
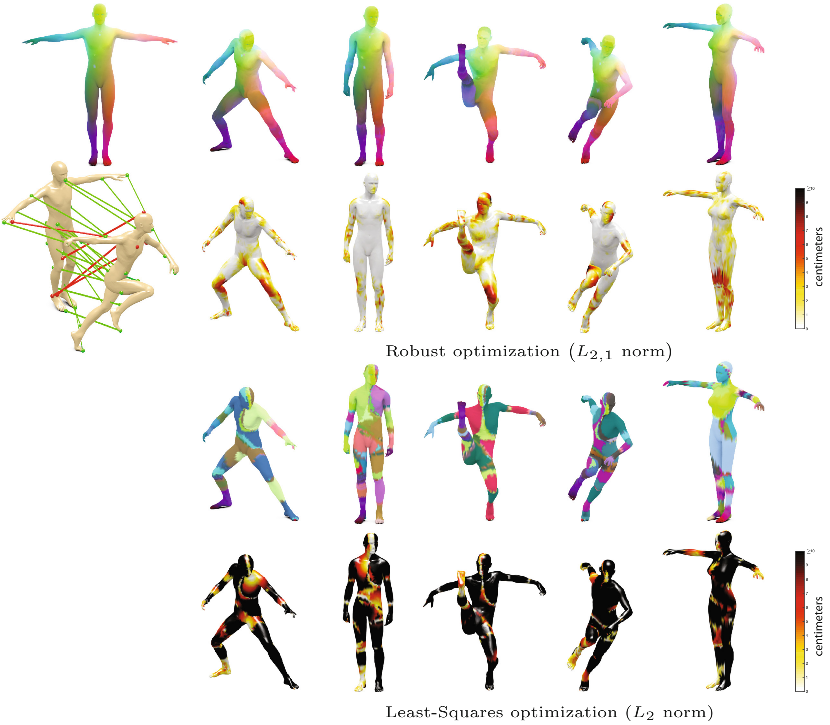


Fig. 4. Functional correspondence problem. Left: evaluation of the functional correspondence obtained using MADMM and least squares. Shown in the percentage of correspondences falling within a geodesic ball of increasing radius w.r.t. the groundtruth correspondence. Right: convergence of MADMM and smoothing method for various values of the smoothing parameter.



Robust optimization ($L_{2,1}$ norm)

Least-Squares optimization (L_2 norm)

Fig. 5. Examples of correspondences obtained with MADMM (top two rows) and least-squares solution (bottom two rows). Rows 1 and 3: similar colors encode corresponding points; rows 2 and 4: color encodes the correspondence error (distance in centimeters to the ground-truth). Leftmost column, 1st row: the reference shape; 2nd row: examples of correspondence between a pair of shapes (outliers are shown in red). (Color figure online)

seeds as correspondence data, contaminated by 16 % outliers. Figure 4 (left) analyzes the resulting correspondence quality using the Princeton protocol [49], plotting the percentage of correspondences falling within a geodesic ball of increasing radius w.r.t. the groundtruth correspondence. For comparison, we show the results of a least-squares solution used in [47] (see Fig. 5). Figure 4 (right) shows the convergence of MADMM in a correspondence problem with $L = 2$ shapes. For comparison, we show the convergence of a smoothed version of the $L_{2,1}$-norm $\|A\|_{2,1} \approx \sum_j \left(\sum_i a_{ij}^2 + \epsilon\right)^{1/2}$ in (8) for various values of the smoothing parameter ϵ.

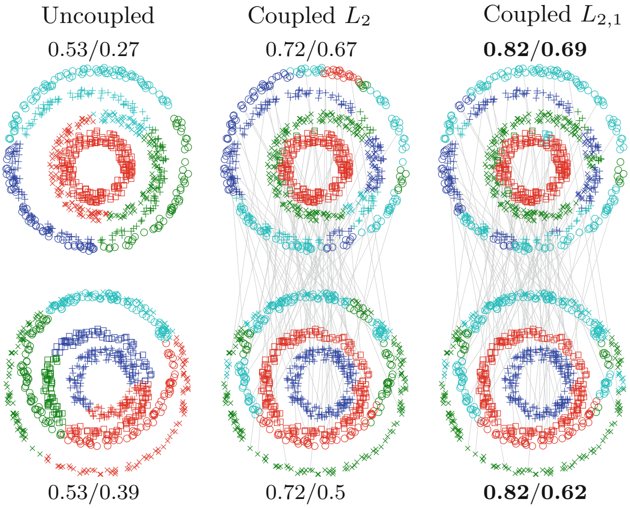


Fig. 6. Clustering of synthetic multimodal datasets *Circles*. Shown is (left to right): spectral clustering applied to each modality independently; clustering results produced by coupled diagonalization methods with L_2 and $L_{2,1}$ norms, respectively. Grey lines depict 10 % of outliers correspondences. Ideally, all markers of each type should have a single color. Numbers show micro-averaged accuracy [50]/normalized NMI [51] (the higher the better).

Figure 6 shows the application of our method for multimodal clustering using the dataset *Circles* from [5], where we introduce 10 % outliers in the correspondence between the modalities data points. Following Eynard *et al.* [5], we use $\hat{\Phi}_i = \Phi_i X_i$ obtained by solving problem (8) as joint multimodal data embedding, and perform spectral clustering [52] in this space. Clustering quality was measured using two standard criteria used in the evaluation of clustering algorithms: the *micro-averaged accuracy* [50] and the *normalized mutual information* (NMI) [51]. The use of the robust $L_{2,1}$ problem formulation (Fig. 6, right) solved with our MADMM outperforms the smooth L_2 version (Fig. 6, center).

4.3 Robust Euclidean Embedding

Problem Setting. Our third problem is an L_1 formulation of the multidimensional scaling (MDS) problem treated in [27] under the name *robust Euclidean embedding* (REE). Let us be given an $n \times n$ matrix D of squared distances. The goal is to find a k-dimensional configuration of points $X \in \mathbb{R}^{n \times k}$ such that the Euclidean distances between them are as close as possible to the given ones. The classical MDS approach employs the duality between Euclidean distance matrices and Gram matrices: a squared Euclidean distance matrix D can be converted into a *similarity matrix* by means of double-centering $B = -\frac{1}{2}HDH$, where $H = I - \frac{1}{n}11^\top$. Conversely, the squared distance matrix is obtained from B by $(\mathrm{dist}(B))_{ij} = b_{ii} + b_{jj} - 2b_{ij}$. The similarity matrix corresponding to a Euclidean distance matrix is positive semi-definite and can be represented as a Gram matrix $B = XX^\top$, where X is the desired embedding. In the case when D is not Euclidean, B acts as a low-rank approximation of the similarity matrix (now not necessarily positive semi-definite) associated with D, leading to the problem

$$\min_{X \in \mathbb{R}^{m \times k}} \|HDH - XX^\top\|_\mathrm{F}^2 \tag{9}$$

known as *classical MDS* or *classical scaling*, which has a closed form solution by means of eigendecomposition of HDH (Fig 7).

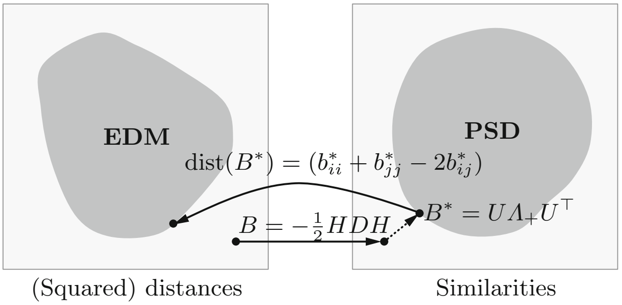


Fig. 7. Illustration of the classical MDS approach and the equivalence between Euclidean distance matrices (EDM) and positive semi-definite (PSD) similarity matrices.

The main disadvantage of classical MDS is the fact that noise in a single entry of the distance matrix D is spread over entire column/row by the double centering transformation. To cope with this problem, Cayton and Dasgupta [27] proposed an L_1 version of the problem,

$$\min_{B \in \mathbb{R}^{n \times n}} \|D - \mathrm{dist}(B)\|_1 \quad \text{s.t.} \quad B \succeq 0,\ \mathrm{rank}(B) \leq k, \tag{10}$$

where the use of the L_1-norm efficiently rejects outliers. The authors proposed two solutions for problem (10): a semi-definite programming (SDP) formulation and a subgradient descent algorithm (the reader is referred to [27] for a detailed description of both methods).

Solution. Here, we consider (10) as a non-smooth optimization of the form (2) on the manifold of fixed-rank positive semi-definite matrices and solve it using MADMM (Algorithm 6). Note that in this case, we have only the non-smooth function g and $f \equiv 0$. The X-step of the MADMM algorithm is manifold optimization of a quadratic function, carried out using two iterations of manifold conjugate gradients solver. The Z-step is performed by shrinkage. In our experiments, all the compared methods were initialized with the classical MDS solution and the value $\rho = 10$ was used for MADMM. The SDP approach was implemented using MATLAB CVX toolbox [53].

```
Input squared distance matrix $D$
Initialize $k \leftarrow 1$, $Z^{(1)} = X^{(1)}$, $U^{(1)} = 0$
repeat
    $B^{(k+1)} = \underset{B \in S_+(n,k)}{\operatorname{argmin}} \ \|\mathrm{dist}(B^{(k+1)}) - Z^{(k)} - D + U^{(k)}\|_\mathrm{F}^2$
    $Z^{(k+1)} = \mathrm{Shrink}_{\frac{1}{\rho}} \left(\mathrm{dist}(B^{(k+1)}) - D + U^{(k)}\right)$
    Update $U^{(k+1)} = U^{(k)} + \mathrm{dist}(B^{(k+1)}) - D - Z^{(k+1)}$
    $k \leftarrow k + 1$
until convergence;
```

Algorithm 6. MADMM method for robust Euclidean embedding.

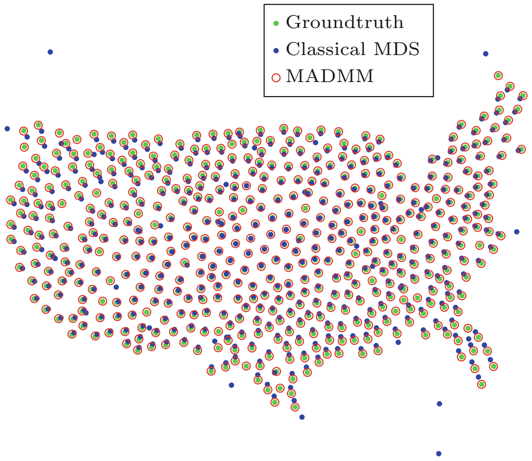


Fig. 8. Embedding of the noisy distances between 500 US cities in the plane using classical MDS (blue) and REE solved using MADMM (red). The distance matrix was contaminated by sparse noise by doubling the distance between some cities. (Color figure online)

Results. Figure 8 shows an example of 2D Euclidean embedding of the distances between 500 US cities, contaminated by sparse noise. The robust embedding is insensitive to such outliers, while the classical MDS result is completely ruined. Figure 9 (right) shows an example of convergence of the proposed MADMM method and the subgradient descent of [27] on the same dataset. We observed that our algorithm outperforms the subgradient method in terms of convergence speed. Furthermore, the subgradient method appears to be very sensitive to the initial step size c; choosing too small a step leads to slower convergence, and if the step is too large the algorithm may fail to converge. Figure 9 (left) studies the scalability of the subgradient-, SDP-, and MADMM-based solutions for the REE problem, plotting the complexity of a single iteration as function of the problem size on random data. Typical number of iterations was of the order of 20 for SDP, 50 for MADMM, and 500 for the subgradient method.

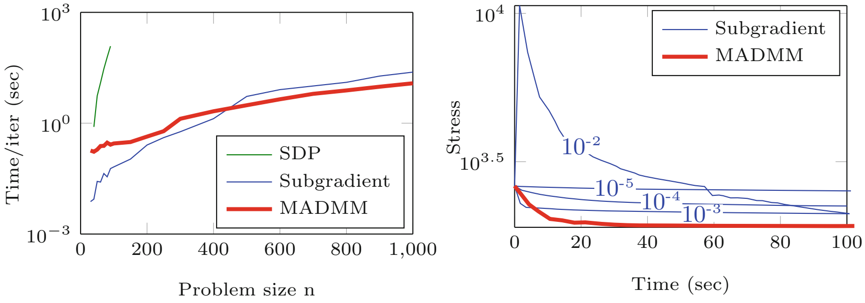


Fig. 9. REE problem. Left: scalability of different algorithms; shown is single iteration complexity as functions of the problem size n using random distance data. SDP did not scale beyond $n = 100$. Right: example of convergence of MADMM and subgradient algorithm of [27] on the US cities problem of size $n = 500$. The subgradient algorithm is very sensitive to the choice of the initial step size c (choosing too large c breaks the convergence, while too small c slows down the convergence).

5 Conclusions

We presented MADMM, a generic algorithm for optimization of non-smooth functions with manifold constraints, and showed that it can be efficiently used in many important problems from the domains of machine learning, computer vision and pattern recognition, and data analysis. Among the key advantages of our method is its remarkable simplicity and lack of parameters to tune - in all our experiments, it worked entirely out-of-the-box. While there exist several solutions for some instances of non-smooth manifold optimization (notably on Stiefel manifolds), MADMM, to the best of our knowledge, is the first generic approach. We believe that MADMM will be very useful in many other applications in the computer vision and pattern recognition community involving manifold optimization. In our experiments, we observed that MADMM converged on par with or better than other methods; a theoretical study of convergence properties is an important future direction. The implementation of the considered problems is at https://github.com/skovnats/madmm.

Acknowledgement. This research was supported by the ERC Starting Grant No. 307047 (COMET).

References

1. Dong, X., Frossard, P., Vandergheynst, P., Nefedov, N.: Clustering on multi-layer graphs via subspace analysis on Grassmann manifolds. Trans. Sig. Process. **62**(4), 905–918 (2014)
2. Keshavan, R.H., Oh, S.: A gradient descent algorithm on the Grassman manifold for matrix completion (2009). arXiv:0910.5260
3. Boufounos, P.T., Baraniuk, R.G.: 1-bit compressive sensing. In: Proceedings of CISS (2008)
4. Kovnatsky, A., Bronstein, M.M., Bronstein, A.M., Glashoff, K., Kimmel, R.: Coupled quasi-harmonic bases. Comput. Graph. Forum **32**(2), 439–448 (2013)
5. Eynard, D., Kovnatsky, A., Bronstein, M.M., Glashoff, K., Bronstein, A.M.: Multimodal manifold analysis using simultaneous diagonalization of Laplacians. Trans. PAMI **37**(12), 2505–2517 (2015)
6. Cucuringu, M., Lipman, Y., Singer, A.: Sensor network localization by eigenvector synchronization over the Euclidean group. ACM Trans. Sensor Netw. **8**(3), 19 (2012)
7. Cucuringu, M., Singer, A., Cowburn, D.: Eigenvector synchronization, graph rigidity and the molecule problem. Inf. Inference **1**(1), 21–67 (2012)
8. Arie-Nachimson, M., Kovalsky, S.Z., Kemelmacher-Shlizerman, I., Singer, A., Basri, R.: Global motion estimation from point matches. In: Proceedings of 3DIMPVT (2012)
9. Zhang, M., Fletcher, P.T.: Probabilistic principal geodesic analysis. In: Proceedings of NIPS (2013)
10. Journée, M., Bach, F., Absil, P.A., Sepulchre, R.: Low-rank optimization on the cone of positive semidefinite matrices. SIAM J. Optimization **20**(5), 2327–2351 (2010)
11. Meyer, G., Bonnabel, S., Sepulchre, R.: Linear regression under fixed-rank constraints: a Riemannian approach. In: Proceedings of ICML (2011)
12. Boumal, N., Absil, P.A.: RTRMC: A Riemannian trust-region method for low-rank matrix completion. In: Procedings of NIPS, pp. 406–414 (2011)
13. Tan, M., Tsang, I.W., Wang, L., Vandereycken, B., Pan, S.J.: Riemannian pursuit for big matrix recovery. In: Proceedings of ICML (2014)
14. Shalit, U., Weinshall, D., Chechik, G.: Online learning in the manifold of low-rank matrices. In: Proceedings of NIPS (2010)
15. Absil, P.A., Gallivan, K.A.: Joint diagonalization on the oblique manifold for independent component analysis. In: Proceedings of ICASSP (2006)
16. Kleinsteuber, M., Shen, H.: Blind source separation with compressively sensed linear mixtures. Sig. Process. Lett. **19**(2), 107–110 (2012)
17. Higham, N.J.: Computing the nearest correlation matrix - a problem from finance. IMA J. Numer. Anal. **22**(3), 329–343 (2002)

18. Smith, S.T.: Optimization techniques on Riemannian manifolds. Fields Inst. Commun. **3**(3), 113–135 (1994)
19. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2009)
20. Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl. **20**(2), 303–353 (1998)
21. Absil, P.A., Baker, C.G., Gallivan, K.A.: Trust-region methods on Riemannian manifolds. Found. Comput. Math. **7**(3), 303–330 (2007)
22. Alvarez, F., Bolte, J., Munier, J.: A unifying local convergence result for Newton's method in Riemannian manifolds. Found. Comput. Math. **8**(2), 197–226 (2008)
23. Boumal, N., Mishra, B., Absil, P.A., Sepulchre, R.: Manopt, a Matlab toolbox for optimization on manifolds. JMLR **15**(1), 1455–1459 (2014)
24. Candès, E., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? J. ACM **58**(3), 11 (2011)
25. Ozoliņš, V., Lai, R., Caflisch, R., Osher, S.: Compressed modes for variational problems in mathematics and physics. PNAS **110**(46), 18368–18373 (2013)
26. Neumann, T., Varanasi, K., Theobalt, C., Magnor, M., Wacker, M.: Compressed manifold modes for mesh processing. Comput. Graphics Forum **33**(5), 35–44 (2014)
27. Cayton, L., Dasgupta, S.: Robust Euclidean embedding. In: Proceedings of ICML (2006)
28. Wang, L., Singer, A.: Exact and stable recovery of rotations for robust synchronization. Information and Inference (2013)
29. Huang, Q., Wang, F., Guibas, L.: Functional map networks for analyzing and exploring large shape collections. ACM Trans. Graphics **33**(4), 36 (2014)
30. Kovnatsky, A., Bronstein, M.M., Bresson, X., Vandergheynst, P.: Functional correspondence by matrix completion. In: Proceedings of CVPR (2015)
31. Chen, X.: Smoothing methods for nonsmooth, nonconvex minimization. Math. Program. **134**(1), 71–99 (2012)
32. Ferreira, O.P., Oliveira, P.R.: Subgradient algorithm on Riemannian manifolds. J. Optimization Theory Appl. **97**(1), 93–104 (1998)
33. Ledyaev, Y., Zhu, Q.: Nonsmooth analysis on smooth manifolds. Trans. AMS **359**(8), 3687–3732 (2007)
34. Borckmans, P.B., Selvan, S.E., Boumal, N., Absil, P.A.: A Riemannian subgradient algorithm for economic dispatch with valve-point effect. J. Comp. Applied Math. **255**, 848–866 (2014)
35. Lai, R., Osher, S.: A splitting method for orthogonality constrained problems. J. Scientific Comput. **58**(2), 431–449 (2014)
36. Rosman, G., Wang, Y., Tai, X., Kimmel, R., Bruckstein, A.M.: Fast regularization of matrix-valued images. In: Proceedings of Efficient Algorithms for Global Optimization Methods in Computer Vision (2011)
37. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximation. Comput. Math. Appl. **2**(1), 17–40 (1976)
38. Boyd, S., Parikk, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**, 1–122 (2010)
39. Hestenes, M.R.: Multiplier and gradient methods. J. Optim. Theory Appl. **4**(5), 303–320 (1969)
40. Powell, M.J.D.: A method for nonlinear constraints in minimization problems. In: Optimization. Academic Press, London, New York (1969)

41. Bertsekas, D.P.: Constrained Optimization and Lagrange Multiplier Methods. Academic Press, New York (1982)
42. Andreani, R., Birgin, E.G., Martínez, J.M., Schuverdt, M.L.: On augmented Lagrangian methods with general lower-level constraints. SIAM J. Optimization **18**(4), 1286–1309 (2007)
43. Attouch, H., Bolte, J., Redont, P., Soubeyran, A.: Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the Kurdyka-Lojasiewicz inequality. Math. Oper. Res. **35**(2), 438–457 (2010)
44. Wang, Y., Wotao, Y., Jinshan, Z.: Global Convergence of ADMM in Nonconvex Nonsmooth Optimization (2015). arXiv:1511.06324
45. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Proceedings of NIPS (2002)
46. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Proceedings of NIPS (2001)
47. Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L.J.: Functional maps: a flexible representation of maps between shapes. ACM Trans. Graphics **31**(4), 1–11 (2012)
48. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Numerical Geometry of Non-rigid Shapes. Springer, New York (2008)
49. Kim, V.G., Lipman, Y., Funkhouser, T.: Blended intrinsic maps. Trans. Graphics **30**, 79 (2011)
50. Bekkerman, R., Jeon, J.: Multi-modal clustering for multimedia collections. In: Proceedings of CVPR (2007)
51. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
52. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Proceedings of NIPS (2001)
53. Grant, M., Boyd, S.: Graph implementations for nonsmooth convex programs. In: Blondel, V., Boyd, S., Kimura, H. (eds.) Recent Advances in Learning and Control. LNCIS, vol. 371, pp. 95–110. Springer, Heidelberg (2008)