

Structured Sparse Subspace Clustering: A Unified Optimization Framework

Chun-Guang Li
SICE, Beijing University of Posts
and Telecommunications

René Vidal
Center for Imaging Science
Johns Hopkins University

Abstract

Subspace clustering refers to the problem of segmenting data drawn from a union of subspaces. State of the art approaches for solving this problem follow a two-stage approach. In the first step, an affinity matrix is learned from the data using sparse or low-rank minimization techniques. In the second step, the segmentation is found by applying spectral clustering to this affinity. While this approach has led to state of the art results in many applications, it is sub-optimal because it does not exploit the fact that the affinity and the segmentation depend on each other. In this paper, we propose a unified optimization framework for learning both the affinity and the segmentation. Our framework is based on expressing each data point as a structured sparse linear combination of all other data points, where the structure is induced by a norm that depends on the unknown segmentation. We show that both the segmentation and the structured sparse representation can be found via a combination of an alternating direction method of multipliers with spectral clustering. Experiments on a synthetic data set, the Hopkins 155 motion segmentation database, and the Extended Yale B data set demonstrate the effectiveness of our approach.

1. Introduction

In many real-world applications, we need to deal with high-dimensional datasets, such as images, videos, text, and more. In practice, such high-dimensional datasets can often be well approximated by multiple low-dimensional subspaces corresponding to multiple classes or categories. For example, the feature point trajectories associated with a rigidly moving object in a video lie in an affine subspace (of dimension up to 4) [38], and face images of a subject under varying illumination lie in a linear subspace (of dimension up to 9) [18]. Therefore, the task, known in the literature as *subspace clustering*, is to segment the data into the corresponding subspaces and has many applications in computer vision, e.g., image representation and compression [19], motion segmentation [8, 36], and temporal video segmentation [43].

Previous Work. The subspace clustering problem has received a lot of attention over the past few years and many methods have been developed, including iterative methods [5, 40, 48, 49, 1], algebraic methods [3, 43, 32, 21], statistical methods [22, 2, 17, 31, 47], and spectral clustering based methods [46, 15, 11, 7, 49, 9, 10, 25, 24, 12, 42, 28, 27, 29] (see [41] for details). Among them, methods based on spectral clustering have become extremely popular. Such methods divide the problem in two steps. In the first one, an affinity matrix is learned from the data. In the second one, spectral clustering is applied to this affinity. Arguably, the first step is the most important, as the success of the spectral clustering algorithm is largely dependent on constructing an informative affinity matrix.

Recent methods for learning the affinity matrix are based on the self-expressiveness model, which states that a point in a union of subspaces can be expressed as a linear combination in terms of other data points, i.e., $X = XZ$, where X is the data matrix and Z is the matrix of coefficients. With corrupted data, this constraint is relaxed to $X = XZ + E$, where E is a matrix of errors. The subspace clustering problem is then formulated as the following optimization problem:

$$\min_{Z, E} \|Z\|_{\kappa} + \lambda \|E\|_{\ell} \text{ s.t. } X = XZ + E, \text{ diag}(Z) = 0, \quad (1)$$

where $\|\cdot\|_{\kappa}$ and $\|\cdot\|_{\ell}$ are two properly chosen norms, $\lambda > 0$ is a tradeoff parameter, and the constraint $\text{diag}(Z) = 0$ is optionally used to rule out the trivial solution of Z being an identity matrix.

The primary difference between different methods lies in the choice of norms for the regularization on Z and/or the noise term E . For example, in Sparse Subspace Clustering (SSC) [9, 10] the ℓ_1 norm is used for $\|\cdot\|_{\kappa}$ as a convex surrogate over the ℓ_0 norm to promote sparseness in Z , and the Frobenius norm and/or the ℓ_1 norm of E is used to handle Gaussian noise and/or outlying entries. In Low-Rank Representation (LRR) [25, 24] and Low-Rank Subspace Clustering (LRSC) [12, 42] the nuclear norm $\|\cdot\|_*$ is adopted as a convex surrogate of the rank function, the $\ell_{2,1}$ norm of E is used to tackle outliers (LRR), and the Frobenius norm and/or the ℓ_1 of E is used to handle noise

and/or outlying entries. In LSR [28], the Frobenius norm of Z is used in order to yield a block diagonal representation matrix Z when subspaces are independent. In Correlation Adaptive Subspace Segmentation (CASS) [27], the Trace-Lasso norm [16] is used for each column of Z to gain a tradeoff effect between ℓ_1 and ℓ_2 norms. Locality or spatial information has also been adopted for weighting or regularization on Z [35][29][20]. Also, a number of variants of these algorithms have been proposed, including Latent LRR [26], Multiple Subspace Recovery (MSR) [30], Subspace Segmentation with Quadratic Programming (SSQP) [45], Spatial Weighted SSC [35], Latent SSC [33], Kernel SSC [34], *etc.* Once the matrix of coefficients Z has been found by any of the methods above, the segmentation of the data is then obtained by applying spectral clustering to an affinity matrix induced from Z , e.g., $|Z| + |Z^\top|$.

While the above methods have been incredibly successful in many applications, an important disadvantage is that they divide the problem into two separate stages:

- Compute the coefficients matrix Z of the self-expressiveness model, using, e.g., SSC, LRR, LRSC, LSR, *etc.*
- Construct an affinity matrix, e.g., $|Z| + |Z^\top|$, and apply a spectral clustering to find the segmentation of the data.

Dividing the problem in two steps is, on the one hand, appealing because the first step can be solved using convex optimization techniques, while the second step can be solved using existing spectral clustering techniques. On the other hand, however, its major disadvantage is that the natural relationship between the affinity matrix and the segmentation of the data is not explicitly captured.

To the best of our knowledge, the only attempt to integrate these two stages into a unified framework is the work of Feng *et al.* [13], who introduce a block-diagonal constraint into the self-expressiveness model. However, this requires precise knowledge of the segmentation, and therefore enforcing exact block-diagonality is not possible with their model.

In this paper, we attempt to integrate the two separate stages into one unified optimization framework. One important observation is that a perfect subspace clustering can often be obtained from an imperfect affinity matrix. In other words, the spectral clustering step can clean up the disturbance in the affinity matrix – which can be viewed as a process of information gain by denoising. Because of this, if we feed back the information gain properly, it may help the self-expressiveness model to yield a better affinity matrix. As shown in Fig. 1, the clustering results can help the self-expressiveness model find an improved affinity matrix and thus boost the final clustering results.

Paper Contributions. In this paper, we propose a subspace structured norm on the self-expression coefficients matrix and turn the problem of subspace clustering into a unified optimization framework. Then, by integrating this norm with the standard ℓ_1 norm, we propose a *subspace structured* ℓ_1 norm and reformulate SSC into a unified optimization framework, termed as Structured Sparse Subspace Clustering (SSSC or $\mathbf{S}^3\mathbf{C}$), in which the separate two stages of computing the sparse representation and applying the spectral clustering are merged together automatically. More specifically, in $\mathbf{S}^3\mathbf{C}$ we use the output of spectral clustering to define a subspace structure matrix, which is feed back to re-weight the representation matrix in the next iteration, and thus unify the two separate stages into a single optimization framework.

Our work has two main contributions.

1. To propose a subspace structured norm on the self-expression coefficients matrix and formulate the subspace clustering problem into a unified optimization framework.
2. To propose a subspace structured ℓ_1 norm and reformulate the sparse subspace clustering into a unified optimization framework, which is solved efficiently by a combination of an alternating direction method of multipliers with spectral clustering.

2. A Unified Optimization Framework for Subspace Clustering

This paper is concerned with the following problem.

Problem 2.1 (Subspace clustering). Let $X \in \mathbb{R}^{n \times N}$ be a real-valued matrix whose columns are drawn from a union of k subspaces of \mathbb{R}^n , $\bigcup_{j=1}^k \{S_j\}$, of dimensions $d_j < \min\{n, N\}$, for $j = 1, \dots, k$. The goal of subspace clustering is to segment the columns of X into their corresponding subspaces.

To begin with, we introduce some additional notation. Let $Q = [\mathbf{q}_1, \dots, \mathbf{q}_k]$ be an $N \times k$ binary matrix indicating the membership of each data point to each subspace. That is, $q_{ij} = 1$ if the i -th column of X lies in subspace S_j and $q_{ij} = 0$ otherwise. We assume that each data point lies in only one subspace, hence if Q is a valid segmentation matrix, we must have $Q\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is the vector of all ones of appropriate dimension. Note that the number of subspaces is equal to k , so we must have that $\text{rank}(Q) = k$. Thus, we define the space of segmentation matrices as

$$\mathcal{Q} = \{Q \in \{0, 1\}^{N \times k} : Q\mathbf{1} = \mathbf{1} \text{ and } \text{rank}(Q) = k\}. \quad (2)$$

2.1. Structured Subspace Clustering: A Unified Framework

Recall from (1) that data in a union of subspaces are self-expressive, that is, each data point in a union of subspaces

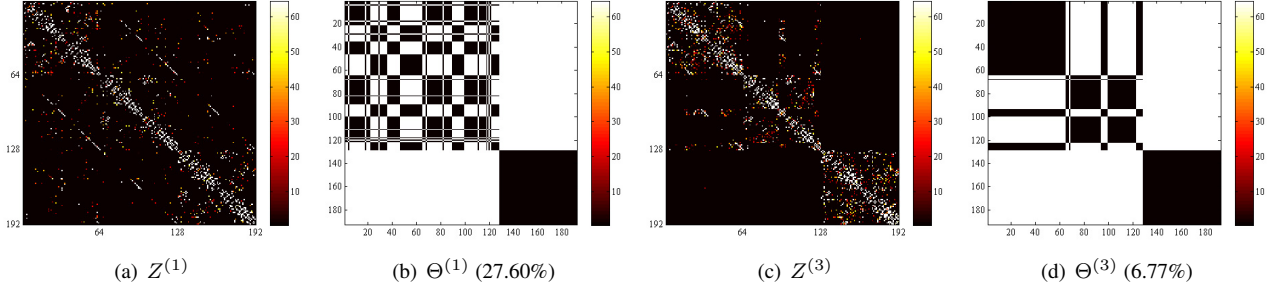


Figure 1. Visualization of matrices $Z^{(t)}$ and $\Theta^{(t)}$ in the first and the third iterations of S^3C algorithm. Note that the first iteration of S^3C is effectively a SSC and hence the images of $Z^{(1)}$ and $\Theta^{(1)}$ in panel (a) and (b) are the representation matrix and the structure matrix of SSC, in which $\Theta^{(1)}$ is computed using the current clustering result. The structure matrix $\Theta^{(t)}$ is used to reweight the computing of the representation matrix $Z^{(t+1)}$ in the next iteration. The images in panel (c) and (d) are the representation matrix $Z^{(t)}$ and structure matrix $\Theta^{(t)}$ of S^3C when converged ($t = 3$). The percentage numbers in bracket are the corresponding clustering errors. For ease of visualization, we computed the entry-wise absolute value and amplified each entry $|Z_{ij}|$ and Θ_{ij} by a factor of 500.

can be expressed as a linear combination of other data points as $X = XZ$ with $\text{diag}(Z) = \mathbf{0}$, where $Z \in \mathbb{R}^{N \times N}$ is a matrix of coefficients whose (i, j) entry $|Z_{ij}|$ captures the similarity between points i and j . The matrix of coefficients is called *subspace structured* provided that $Z_{ij} \neq 0$ only if points i and j lie in the same subspace.

In existing approaches [10, 25, 28], one exploits the self-expressiveness property by adding some penalty term on Z , e.g., ℓ_1 [9, 10], $\|\cdot\|_*$ [25, 24], $\|\cdot\|_F$ [28], or even a data dependent norm [27]. However, these approaches do not exploit the fact that the representation matrix Z and the segmentation matrix Q both try to capture the segmentation of the data. Indeed, recall that the approach in (1) computes the segmentation Q by applying spectral clustering to the similarity matrix $|Z| + |Z^\top|$.

For the time being, imagine that the exact segmentation matrix Q was known. In order to capture the fact that the zero patterns of Z and Q are related, observe that when $Z_{ij} \neq 0$ points i and j are in the same subspace, hence we must have $\mathbf{q}^{(i)} = \mathbf{q}^{(j)}$, where $\mathbf{q}^{(i)}$ and $\mathbf{q}^{(j)}$ are the i -th and j -th row of matrix Q , respectively. Therefore, we can quantify the disagreement between Z and Q by using the following *subspace structured* measure of Z :

$$\|\Theta \odot Z\|_0 = \sum_{i,j: \mathbf{q}^{(i)} \neq \mathbf{q}^{(j)}} \mathbb{I}_{\{|Z_{ij}| \neq 0\}} \quad (3)$$

where $\Theta_{ij} = \frac{1}{2} \|\mathbf{q}^{(i)} - \mathbf{q}^{(j)}\|^2$, the operator \odot indicates the Hadamard product (i.e., element-wise product) and \mathbb{I} is an indicator function. Since $\Theta_{ij} \in \{0, 1\}$, the cost in (3) effectively counts the number of nonzero entries in Z when data point i and j are in different subspaces. Unfortunately the subspace structured measure of Z is combinatorial in nature and hence leads to an NP-hard problem.

Instead, we propose a relaxed counterpart, called the *subspace structured* norm of Z with respect to (w.r.t.) Q ,

denoted as $\|Z\|_Q$, which is defined as follows:

$$\|Z\|_Q \doteq \sum_{i,j} |Z_{ij}| \left(\frac{1}{2} \|\mathbf{q}^{(i)} - \mathbf{q}^{(j)}\|^2 \right) = \|\Theta \odot Z\|_1. \quad (4)$$

Note that when the *subspace detection property* [37] holds, i.e., $Z_{ij} = 0$ whenever points i and j lie in different subspace, then the subspace structured norm $\|Z\|_Q$ vanishes; otherwise, it is positive.

In practice, of course, the true segmentation matrix Q is unknown. Nevertheless, $\|Z\|_Q$ still provides a useful measure of the agreement between the segmentation matrix Q and the representation matrix Z . Therefore, a straightforward approach to subspace clustering is to use the subspace structured norm $\|Z\|_Q$ directly to replace the first term $\|Z\|_\kappa$ in problem (1) and optimize over Z , E , and Q simultaneously. By doing so, we have a unified optimization framework, which ensures consistency between the representation coefficients and the subspace segmentation by solving the following optimization problem:

$$\begin{aligned} \min_{Z, E, Q} \quad & \|Z\|_Q + \lambda \|E\|_\ell \\ \text{s.t.} \quad & X = XZ + E, \text{diag}(Z) = \mathbf{0}, Q \in \mathcal{Q}. \end{aligned} \quad (5)$$

The optimization problem in (5) can be solved by alternatively solving (Z, E) and Q . Given Q , the problem is a convex program for (Z, E) which can be solve efficiently using the alternating direction method of multipliers (ADMM) [4][23]. Given (Z, E) , the solution for Q can be computed approximately by spectral clustering. To see this, note that the $\|Z\|_Q$ can be rewritten as:

$$\begin{aligned} \|Z\|_Q &= \sum_{i,j} |Z_{ij}| \left(\frac{1}{2} \|\mathbf{q}^{(i)} - \mathbf{q}^{(j)}\|^2 \right) \\ &= \frac{1}{2} \sum_{i,j} A_{i,j} \|\mathbf{q}^{(i)} - \mathbf{q}^{(j)}\|_2^2 \\ &= \text{trace}(Q^\top \bar{L} Q), \end{aligned} \quad (6)$$

where $A_{ij} = \frac{1}{2}(|Z_{ij}| + |Z_{ji}|)$ measures the similarity of points i and j , $\bar{L} = \bar{D} - A$ is a graph Laplacian, and \bar{D} is a diagonal matrix whose diagonal entries are $\bar{D}_{jj} = \sum_i A_{ij}$. Consequently, given Z , we can find Q by solving the problem:

$$\min_Q \text{trace}(Q^\top \bar{L} Q) \quad \text{s.t.} \quad Q \in \mathcal{Q}, \quad (7)$$

which is the problem solved approximately by spectral clustering by relaxing the constraint $Q \in \mathcal{Q}$ to $Q^\top Q = I$.

Although the formulation in (5) is appealing since it formulates the two separate stages in self-expression model based spectral clustering methods into a unified optimization framework, it suffers from two critical shortcomings: a) it depends strongly on initialization, and b) it depends tightly upon solving the subspace segmentation matrix Q which is combinatorial problem. Note that:

- When Q is initialized arbitrarily as a segmentation matrix in \mathcal{Q} , the iterations of the optimization problem will be “trapped” within the initialized support of Θ which is defined by the initial Q .
- If Q is initialized from an infeasible solution, which corresponds to treating each single data point as one subspace, then the first iteration for solving Z is equivalent to a standard SSC.¹ Unfortunately, after solving one iteration of the problem, we then return to the previous issue, where we become trapped within the first estimated clustering.

To remedy the drawbacks of the unified optimization framework mentioned above, we need to incorporate additional structure into the subspace structured norm $\|Z\|_Q$.

2.2. Structured Sparse Subspace Clustering (S³C)

Recall that in existing work, the ℓ_1 norm $\|Z\|_1$ used in SSC, the nuclear norm $\|Z\|_*$ used in LRR, the Frobenius norm $\|Z\|_F$ used in LSR, or the TraceLasso norm used in CASS are all powerful means to detect subspaces. While any of them could potentially be incorporated into the subspace structured norm, here we adopt the ℓ_1 norm and define a *subspace structured* ℓ_1 norm of Z as follows:

$$\begin{aligned} \|Z\|_{1,Q} &\doteq \|Z\|_1 + \alpha \|\Theta \odot Z\|_1 \\ &= \|(\mathbf{1}\mathbf{1}^\top + \alpha\Theta) \odot Z\|_1 \\ &= \sum_{i,j} |Z_{ij}| (1 + \frac{\alpha}{2} \|\mathbf{q}^{(i)} - \mathbf{q}^{(j)}\|^2), \end{aligned} \quad (8)$$

where $\alpha > 0$ is a tradeoff parameter. Clearly, the first term is the standard ℓ_1 norm used in SSC. Therefore the subspace structured ℓ_1 norm can be viewed as an ℓ_1 norm augmented

¹In this case, since $\Theta = \mathbf{1}\mathbf{1}^\top - \text{diag}(\mathbf{1})$, we have $\|Z\|_Q = \|Z\|_1 - \text{diag}(Z)$.

by an extra penalty on Z_{ij} when points i and j are in different subspaces according to the segmentation matrix Q . The reasons we prefer to use the ℓ_1 norm are two fold:

- Since both the ℓ_1 norm and the subspace structured norm (w.r.t. Q) are based on the ℓ_1 norm, this leads to a combined norm that also has the structure of the ℓ_1 norm. We will see later that this will facilitate updating the coefficients of Z .
- The second reason to use the ℓ_1 norm is the great theoretical guarantees for correctness enjoyed by SSC, which is applicable to detect subspace even when subspaces are overlapping [37].

Equipped with the *subspace structured* ℓ_1 norm of Z , we can reformulate the unified optimization framework in (5) for subspace clustering as follows:

$$\begin{aligned} \min_{Z,E,Q} \|Z\|_{1,Q} + \lambda \|E\|_\ell \\ \text{s.t.} \quad X = XZ + E, \text{diag}(Z) = \mathbf{0}, Q \in \mathcal{Q}. \end{aligned} \quad (9)$$

where the norm $\|\cdot\|_\ell$ on the error term E depends upon the prior knowledge about the pattern of noise or corruptions.² We call problem (9) *Structured Sparse Subspace Clustering* (SSSC or S³C).

Discussions. Notice that the S³C framework in (9) generalizes SSC because, instead of first solving for a sparse representation to find Z and then applying spectral clustering to the affinity $|Z| + |Z^\top|$ to obtain the segmentation, in (9) we simultaneously search for the sparse representation Z and the segmentation Q . Our S³C differs from the reweighted ℓ_1 minimization [6] in that what we use to reweight is not Z itself but rather a subspace structure matrix Θ . Compared to [13], which adds an extra block-diagonal constraint into the expressiveness model, whereas our framework encourages consistency between the representation matrix and the estimated segmentation matrix. It should also be noted that our subspace structured norm can be used in conjunction with other self-expressiveness based methods [25][28][27], and can also be combined with other weighted methods [35] or methods with extra regularization [20].

3. An Alternating Minimization Algorithm for Structured Sparse Subspace Clustering

In this section, we propose a solution to the optimization problem in (9) based on solving the following two subproblems alternatively:

²For example, the Frobenius norm will be used if the data are contaminated with dense noise; the ℓ_1 norm will be used if the data are contaminated with sparse corruptions; the $\ell_{2,1}$ norm will be used if the data are contaminated with gross corruptions over a few columns; or the combination of these norms will be used for mixed patterns of noise and corruptions.

Algorithm 1 (ADMM for solving problem (11))

Input: Data matrix X , Θ_0 , λ , and α .

Initialize: $\Theta = \Theta_0$, $E = \mathbf{0}$, $C = Z = Z_0$, $Y^{(1)} = \mathbf{0}$, $Y^{(2)} = \mathbf{0}$, $\epsilon = 10^{-6}$, $\rho = 1.1$

while not converged **do**

 Update Z_t , C_t , and E_t ;

 Update $Y_t^{(1)}$ and $Y_t^{(2)}$;

 Update $\mu_{t+1} \leftarrow \rho\mu_t$;

 Check the convergence condition $\|X - XC - E\|_\infty < \epsilon$; if not converged, then set $t \leftarrow t + 1$.

end while

Output: Z_{t+1} and E_{t+1}

1. Find Z and E given Q by solving a weighted sparse representation problem.
2. Find Q given Z and E by spectral clustering.

3.1. Subspace Structured Sparse Representation

Given the segmentation matrix Q (or the structure matrix Θ), we solve for Z and E by solving the following structured sparse representation problem

$$\begin{aligned} \min_{Z, E} \quad & \|Z\|_{1,Q} + \lambda \|E\|_\ell \\ \text{s.t.} \quad & X = XZ + E, \text{diag}(Z) = \mathbf{0}, \end{aligned} \quad (10)$$

which is equivalent to the following problem

$$\begin{aligned} \min_{Z, C, E} \quad & \|Z\|_{1,Q} + \lambda \|E\|_\ell \\ \text{s.t.} \quad & X = XC + E, C = Z - \text{diag}(Z). \end{aligned} \quad (11)$$

We solve this problem using the Alternating Direction Method of Multipliers (ADMM) [23],[4]. The augmented Lagrangian is given by:

$$\begin{aligned} \mathcal{L}(Z, C, E, Y^{(1)}, Y^{(2)}) \\ = & \|Z\|_{1,Q} + \lambda \|E\|_\ell + \langle Y^{(1)}, X - XC - E \rangle \\ & + \langle Y^{(2)}, C - Z + \text{diag}(Z) \rangle \\ & + \frac{\mu}{2} (\|X - XC - E\|_F^2 + \|C - Z + \text{diag}(Z)\|_F^2), \end{aligned} \quad (12)$$

where $Y^{(1)}$ and $Y^{(2)}$ are matrices of Lagrange multipliers, and $\mu > 0$ is a parameter. To find a saddle point for \mathcal{L} , we update each of Z , C , E , $Y^{(1)}$, and $Y^{(2)}$ alternatively while keeping the other variables are fixed.

Update for Z . We update Z by solving the following problem:

$$Z_{t+1} = \arg \min_Z \frac{1}{\mu_t} \|Z\|_{1,Q} + \frac{1}{2} \|Z - \text{diag}(Z) - U_t\|_F^2, \quad (13)$$

where $\|Z\|_{1,Q} = \|Z\|_1 + \alpha \|\Theta \odot Z\|_1$ and $U_t = C_t + \frac{1}{\mu_t} Y_t^{(2)}$. The closed-form solution for Z is given as

$$Z_{t+1} = \tilde{Z}_{t+1} - \text{diag}(\tilde{Z}_{t+1}), \quad (14)$$

where the (i, j) entry of \tilde{Z} is given by

$$\tilde{Z}_{t+1}^{ij} = \mathcal{S}_{\frac{1}{\mu_t}(1+\alpha\Theta_{ij})}(U_t^{ij}). \quad (15)$$

where $\mathcal{S}_\tau(\cdot)$ is the shrinkage thresholding operator. Note that the subspace structured ℓ_1 norm causes a minor change to the algorithm used to compute Z from the standard SSC – because of the homogeneity in the two terms. Namely, rather than soft-thresholding all the entries of matrix U_t with a constant value, we threshold the entries of matrix U_t with different values that depend on Θ_{ij} .

Update for C . We update C by solving the following problem:

$$\begin{aligned} C_{t+1} = \arg \min_C \quad & \langle Y_t^{(1)}, X - XC - E_t \rangle \\ & + \langle Y_t^{(2)}, C - Z_t + \text{diag}(Z_t) \rangle \\ & + \frac{\mu_t}{2} (\|X - XC - E_t\|_F^2 + \|C - Z_t + \text{diag}(Z_t)\|_F^2), \end{aligned} \quad (16)$$

whose solution is given by

$$\begin{aligned} C_{t+1} = & (X^\top X + \mathbf{I})^{-1} [X^\top (X - E_t - \frac{1}{\mu_t} Y_t^{(1)}) \\ & + Z_t - \text{diag}(Z_t) - \frac{1}{\mu_t} Y_t^{(2)}]. \end{aligned} \quad (17)$$

Update for E . While other variables are fixed, we update E as follows:

$$E_{t+1} = \arg \min_E \frac{\lambda}{\mu_t} \|E\|_\ell + \frac{1}{2} \|E - V_t\|_F^2 \quad (18)$$

where $V_t = X - XC_{t+1} + \frac{1}{\mu_t} Y_t^{(1)}$. If we use the ℓ_1 norm for E , then

$$E_{t+1} = \mathcal{S}_{\frac{\lambda}{\mu_t}}(V_t). \quad (19)$$

Update for $Y^{(1)}$ and $Y^{(2)}$. The update for the Lagrange multipliers is a simple gradient ascent step

$$\begin{aligned} Y_{t+1}^{(1)} &= Y_t^{(1)} + \mu_t (X - XC_{t+1} - E_{t+1}), \\ Y_{t+1}^{(2)} &= Y_t^{(2)} + \mu_t (C_{t+1} - Z_{t+1} + \text{diag}(Z_{t+1})). \end{aligned} \quad (20)$$

For clarity, we summarize the ADMM algorithm for solving problem (11) in Algorithm 1. For the details of the derivation, we refer the readers to [23, 4].

Algorithm 2 (S^3C)

Input: Data matrix X and the number of subspaces k

Initialize: $\Theta = \mathbf{0}$, λ , α

while not converged **do**

 Given Q , solve problem (10) via Alg. 1 to obtain (Z, E) ;

 Given (Z, E) , solve problem (21) via spectral clustering to obtain Q ;

end while

Output: Segmentation matrix Q

3.2. Spectral Clustering

Given Z and E , problem (9) reduces to the following problem:

$$\min_Q \|Z\|_{1,Q} \quad \text{s.t. } Q \in \mathcal{Q}. \quad (21)$$

Recall that $\|Z\|_{1,Q} = \|Z\|_1 + \alpha \|\Theta \odot Z\|_1$ and observe that the first term in $\|Z\|_{1,Q}$ is not a function of Q , hence it can be dropped. Consequently problem (21) reduces to problem (7), which can be approximately solved with spectral clustering [44]. More specifically, by relaxing the constraint that $Q \in \mathcal{Q}$ to requiring that Q be an orthogonal matrix, i.e., $Q^\top Q = \mathbf{I}$, we obtain:

$$\min_{Q \in \mathbb{R}^{N \times k}} \alpha \text{trace}(Q^\top \bar{L} Q) \quad \text{s.t. } Q^\top Q = \mathbf{I}. \quad (22)$$

The solution to this relaxed problem can be found efficiently by eigenvalue decomposition. In particular, the columns of Q are given by the eigenvectors of \bar{L} associated with the smallest k eigenvalues. The rows of Q are then used as input to the k -means algorithm, which produces a clustering of the rows of Q that can be used to produce a binary matrix $Q \in \{0, 1\}^{N \times k}$ such that $Q\mathbf{1} = \mathbf{1}$.

3.3. Algorithm Summary

For clarity, we summarize the whole scheme to solving problem (9) in Algorithm 2. The algorithm alternates between solving for the matrices of sparse coefficients and the error (Z, E) given the segmentation Q using Algorithm 1 and solving for Q given (Z, E) using spectral clustering. While the problem solved by Algorithm 1 is a convex problem, there is no guarantee that the Algorithm 2 will converge to a global or local optimum because the solution for Q given (Z, E) is obtained in an approximate manner by relaxing the objective function. Nonetheless, our experiments show that the algorithm does converge in practice for appropriate settings of the parameters.

Stopping Criterion. We terminate Algorithm 2 by checking the following condition

$$\|\Theta_{T+1} - \Theta_T\|_\infty < 1, \quad (23)$$

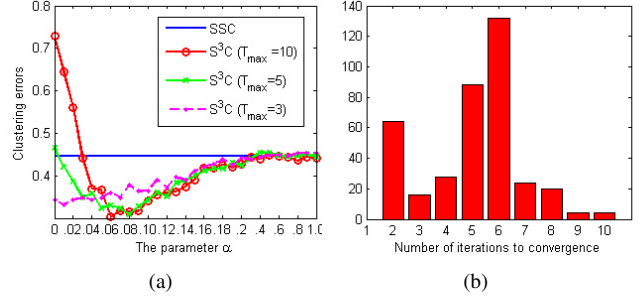


Figure 2. Experimental Results on Synthetic Data Set. (a): The clustering errors as a function of α . (b): The histogram of numbers of iterations where $\alpha = 0.1$ and $T_{max} = 10$.

where $T = 1, 2, \dots$ is the iteration index.

Parameter Setting. In S^3C the parameters λ and α need to be set properly as in SSC [10]. Nevertheless, we observed that rather than using a fixed α to balance the two terms in the subspace structured ℓ_1 norm $\|Z\|_{1,Q}$, the convergence could be improved by, e.g., using $\alpha \leftarrow \nu\alpha$ or using $\nu^{1-T}\|Z\|_1 + \alpha\nu^{T-1}\|Z\|_Q$ in Algorithm 2, where $\nu > 1$ and $T = 1, 2, \dots$ is the iteration index.

4. Experiments

In this section, we evaluate the proposed S^3C approach on a synthetic data set, a motion segmentation data set, and a face clustering data set to validate its effectiveness.

Experimental Setup. Since that our S^3C is a generalization of the standard SSC [10], we keep all settings in S^3C the same as in SSC and thus the first iteration of S^3C is equivalent to a standard SSC. The S^3C specific parameters are set using $\nu^{1-T}\|Z\|_1 + \alpha\nu^{T-1}\|Z\|_Q$ where $\alpha = 0.1$, $\nu = 1.2$, and $T_{max} = 10$ by default. The λ and the ADMM parameters are kept the same for both S^3C and SSC. For each set of experiments, the average and median of subspace clustering error are recorded. The error (ERR) of subspace clustering is calculated by

$$ERR(\mathbf{a}, \hat{\mathbf{a}}) = 1 - \max_{\pi} \frac{1}{N} \sum_{i=1}^N 1_{\{\pi(\mathbf{a}) = \hat{\mathbf{a}}\}} \quad (24)$$

where $\mathbf{a}, \hat{\mathbf{a}} \in \{1, \dots, k\}^N$ are the original and estimated assignments of the columns in X to the k subspaces, and the maximum is with respect to all permutations

$$\pi : \{1, \dots, k\}^N \rightarrow \{1, \dots, k\}^N. \quad (25)$$

For the motion segmentation problem, we consider the Hopkins 155 database [39], which consists of 155 video sequences with 2 or 3 motions in each video corresponding to 2 or 3 low-dimensional subspaces. For the face clustering problem, we consider the Extended Yale B data set [14],

Corruptions (%)	0	10%	20%	30%	40%	50 %	60%	70%	80 %	90%
SSC	1.43	1.93	2.17	4.27	16.87	32.50	54.47	62.43	68.87	73.77
Our S³C	0.30	0.33	0.90	2.97	10.70	23.67	50.50	60.70	67.97	73.33

Table 1. Clustering Errors on Synthetic Data Set. The best results are in bold font.

which consists of face images of 38 subjects, where the face images of each subject correspond to a low-dimensional subspace.

4.1. Experiments on Synthetic Data

Data Preparing. We construct k linear subspaces $\{S_j\}_{j=1}^k \subset \mathbb{R}^n$ whose bases $\{U_j\}_{j=1}^k$ are the left singular matrixes computed from a random matrix $R_j \in \mathbb{R}^{n \times n}$. We sample N_j data points from each subspace $j = 1, \dots, k$ as $X_j = U_j Y_j$, where the entries of $Y_j \in \mathbb{R}^{d \times N_j}$ are i.i.d. samples from a standard Gaussian. Note that by doing so, the k linear subspaces are not necessarily orthogonal to each other. We then corrupt a certain percentage $p = 10 - 90\%$ of entries uniformly at random, e.g., for a column x chosen to corrupt, its observed vector is added by Gaussian noise with zero mean and variance $0.3\|x\|$. We repeat each experiment for 20 trials. In our experiments, we set $n = 100$, $d = 5$, $N_j = 10$, and $k = 15$.

Experimental results are presented in Table 1. Notice that our S³C algorithm consistently outperforms SSC. To gain more insights into the sensitivity of S³C to the parameter α , in Fig. 2 (a) we show the results of varying α over $[0, 1]$ while holding the corruption level fixed at 55%. As could be observed that, our method works well for $\alpha \in [0.03, 0.30]$. In Fig. 2 (b), we show the histogram of the numbers of iterations for S³C to converge with $\alpha = 0.1$ and $T_{max} = 10$. As could be observed that, our S³C converges in $2 \sim 6$ iterations on average. For the average time cost, SSC takes 1.46 seconds whereas our S³C takes 8.05 seconds which is roughly 5.5 times of SSC’s on the synthetic data sets.

4.2. Data Visualization

To show the effect of the subspace structured ℓ_1 norm in S³C intuitively, we take a subset of images of three subjects from the Extended Yale B data set, perform S³C, and then visualize the representation matrix $Z^{(t)}$ and the subspace structure matrix $\Theta^{(t)}$, which are taken in the t -th iteration in S³C. Note that the first iteration of S³C is effectively a SSC since we use the same parameters. The visualization results are displayed in Fig. 1.

We observe from Fig. 1 (a) that the structured sparse representation matrix $Z^{(1)}$ – which is effectively the sparse representation matrix of SSC – is not a clearly block-diagonal and leads to a degenerated clustering result as shown by the subspace structure matrix $\Theta^{(1)}$ in Fig. 1 (b). In the third iteration ($t = 3$), S³C yields a much better representation matrix $Z^{(3)}$, as shown in Fig. 1 (c), and hence produces



Figure 3. Example frames from videos in the Hopkins 155 [39].

a significantly improved clustering result as shown by the subspace structure matrix $\Theta^{(3)}$ in Fig. 1 (d). While S³C in this case did not yield a perfect block-diagonal representation matrix, the improvements still reduce the clustering error significantly – 27.60% vs. 6.77%.

4.3. Experiments on Hopkins 155 Database

Motion segmentation refers to the problem of segmenting a video sequence with multiple rigidly moving objects into multiple spatiotemporal regions that correspond to the different motions in the scene (see Fig. 3). This problem is often solved by first extracting and tracking the spatial positions of a set of N feature points $\mathbf{x}_{fi} \in \mathbb{R}^2$ through each frame $f = 1, \dots, F$ of the video, and then clustering these feature points according to each one of the motions. Under the affine projection model, a feature point trajectory is formed by stacking the feature points \mathbf{x}_{fi} in the video as $\mathbf{y}_i \doteq [\mathbf{x}_{1i}^\top, \mathbf{x}_{2i}^\top, \dots, \mathbf{x}_{Fi}^\top]^\top \in \mathbb{R}^{2F}$. Since the trajectories associated with a single rigid motion lie in an affine subspace of \mathbb{R}^{2F} of dimension at most 4 [38], the trajectories of k rigid motions lie in a union of k low-dimensional subspaces of \mathbb{R}^{2F} . Therefore, the multi-view affine motion segmentation problem reduces to the subspace clustering problem.

We compare our S³C algorithm with SSC [10], LSA [46], LRR [25], LSR [28], BDSSC [13] and BDLRR [13] on the Hopkins 155 motion segmentation data set [39] for the multi-view affine motion segmentation without other postprocessing. Experimental results are presented in Table 2. Note that from the results in Table 2, the S³C is again the best performing method; however, due to the fact that the Hopkins 155 database has a relatively low noise level, the improvement in performance over SSC is relatively minor.

4.4. Experiments on Extended Yale B Data Set

The Extended Yale Database B [14] contains 2,414 frontal face images of 38 subjects, with approximately 64 frontal face images per subject taken under different illumination conditions. Given the face images of multiple subjects acquired under a fixed pose and varying illumination, we consider the problem of clustering the images according

Methods		LSA	LRR	BDLRR	LSR1	LSR2	BDSSC	SSC	Our S ³ C
2 motions	Ave.	3.62	3.76	3.70	2.20	2.22	2.29	1.95	1.94
	ERR(%) Med.	0.51	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3 motions	Ave.	7.67	9.92	6.49	7.13	7.18	4.95	4.94	4.92
	ERR(%) Med.	1.27	1.42	1.20	2.40	2.40	0.91	0.89	0.89
Total	Ave.	4.53	5.15	4.33	3.31	3.34	2.89	2.63	2.61
	ERR(%) Med.	0.63	0.00	0.00	0.22	0.23	0.00	0.00	0.00

Table 2. Motion Segmentation Errors on Hopkins 155 Database. The best results are in bold font.

No. subjects ERR (%)	2		3		5		8		10	
	Average	Median	Average	Median	Average	Median	Average	Median	Average	Median
LRR	6.74	7.03	9.30	9.90	13.94	14.38	25.61	24.80	29.53	30.00
LSR1	6.72	7.03	9.25	9.90	13.87	14.22	25.98	25.10	28.33	30.00
LSR2	6.74	7.03	9.29	9.90	13.91	14.38	25.52	24.80	30.73	33.59
CASS	10.95	6.25	13.94	7.81	21.25	18.91	29.58	29.20	32.08	35.31
LRSC [42]	3.15	2.34	4.71	4.17	13.06	8.44	26.83	28.71	35.89	34.84
BDSSC [13]	3.90	-	17.70	-	27.50	-	33.20	-	39.53	-
BDLRR [13]	3.91	-	10.02	-	12.97	-	27.70	-	30.84	-
LatLRR	2.54	0.78	4.21	2.60	6.90	5.63	14.34	10.06	22.92	23.59
SSC	1.87	0.00	3.35	0.78	4.32	2.81	5.99	4.49	7.29	5.47
S ³ C [†]	1.43	0.00	3.09	0.52	4.08	2.19	4.84	4.10	6.09	5.16
S ³ C [‡]	1.40	0.00	3.08	0.52	3.83	1.88	4.45	3.52	5.42	4.53
S ³ C	1.27	0.00	2.71	0.52	3.41	1.25	4.15	2.93	5.16	4.22

Table 3. Clustering Errors on Extended Yale B Data Set. The best results are in bold font. In S³C[†] we fix α as 0.1 whereas in S³C[‡] we use $\alpha \leftarrow \nu\alpha$.

to their subjects. It has been shown that, under the Lambertian assumption, the images of a subject with a fixed pose and varying illumination lie close to a linear subspace of dimension 9 [18]. Thus, the collection of face images of multiple subjects lie close to a union of 9-dimensional subspaces. It should be noted that the Extended Yale B dataset is more challenging for subspace segmentation than the Hopkins 155 database due to heavy corruptions in the data.

We compare the clustering errors of our proposed S³C algorithm with SSC [10], LRR [25], LatLRR [26], LRSC [42], and other recently proposed algorithms, including LSR [28], CASS [27], and BDSSC [13] and BDLRR [13]. In our experiments, we follow the protocol introduced in [10]: a) each image is down-sampled to be 48×42 pixels and then vectorized to a 2,016-dimensional data point; b) the 38 subjects are then divided into 4 groups – subjects 1-10, 11-20, 21-30, and 31-38. We perform experiments using all choices of $n \in \{2, 3, 5, 8, 10\}$ subjects in each of the first three groups and use all choices of $n \in \{2, 3, 5, 8\}$ from the last group. For S³C, SSC, LatLRR, and LRSC, we use the 2,016-dimensional vectors as inputs. For LatLRR and LRSC, we cite the results reported in [10] and [42], respectively. For LRR, LSR, and CASS, we use the procedure reported in [28]: use standard PCA to reduce the 2,016-dimensional data to $6k$ -dimensional data for $k \in \{2, 3, 5, 8, 10\}$. For BDSSC, and BDLRR, we directly cite the best results in their papers, which are based

on $9k$ -dimensional data. The full experimental results are presented in Table 3. Again, we observe that S³C yields the lowest clustering errors across all experimental conditions.

5. Conclusion

We described a unified optimization framework for the problem of subspace clustering. Moreover by introducing a subspace structured ℓ_1 norm, we formulated the sparse subspace clustering algorithm into a unified optimization framework, termed as Structured Sparse Subspace Clustering (S³C), in which the separate two stages of computing the sparse representation and applying the spectral clustering were elegantly merged together. We solved this problem efficiently via a combination of an alternating direction method of multipliers with spectral clustering. Experiments on a synthetic data set, the Hopkins 155 motion segmentation database, and the Extended Yale B data set demonstrated the effectiveness of our method.

Acknowledgment

C.-G. Li was partially supported by National Natural Science Foundation of China under Grant Nos. 61175011 and 61273217, and Scientific Research Foundation for the Returned Overseas Chinese Scholars, Ministry of Education of China. R. Vidal was supported by the National Science Foundation under Grant No. 1447822.

References

- [1] P. Agarwal and N. Mustafa. k-means projective clustering. In *ACM Symposium on Principles of database systems*, 2004. 1
- [2] C. Archambeau, N. Delannay, and M. Verleysen. Mixtures of robust probabilistic principal component analyzers. *Neurocomputing*, 71(7–9):1274–1282, 2008. 1
- [3] T. Boult and L. Brown. Factorization-based segmentation of motions. In *IEEE Workshop on Motion Understanding*, pages 179–186, 1991. 1
- [4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010. 3, 5
- [5] P. S. Bradley and O. L. Mangasarian. k-plane clustering. *Journal of Global Optimization*, 16(1):23–32, 2000. 1
- [6] E. Candès, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5):877–905, 2008. 4
- [7] G. Chen and G. Lerman. Spectral curvature clustering (SC-C). *International Journal of Computer Vision*, 81(3):317–330, 2009. 1
- [8] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998. 1
- [9] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 1, 3
- [10] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013. 1, 3, 6, 7, 8
- [11] Z. Fan, J. Zhou, and Y. Wu. Multibody grouping by inference of multiple subspaces from high-dimensional data using oriented-frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):91–105, 2006. 1
- [12] P. Favaro, R. Vidal, and A. Ravichandran. A closed form solution to robust subspace estimation and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 1
- [13] J. Feng, Z. Lin, H. Xu, and S. Yan. Robust subspace segmentation with block-diagonal prior. In *CVPR*, 2014. 2, 4, 7, 8
- [14] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001. 6, 7
- [15] A. Goh and R. Vidal. Segmenting motions of different types by unsupervised manifold clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 1
- [16] E. Grave, G. Obozinski, and F. Bach. Trace lasso: a trace norm regularization for correlated designs. In *NIPS*, 2011. 2
- [17] A. Gruber and Y. Weiss. Multibody factorization with uncertainty and missing data using the em algorithm. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 707–714, 2004. 1
- [18] J. Ho, M. H. Yang, J. Lim, K. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003. 1, 8
- [19] W. Hong, J. Wright, K. Huang, and Y. Ma. Multi-scale hybrid linear models for lossy image representation. *IEEE Trans. on Image Processing*, 15(12):3655–3671, 2006. 1
- [20] H. Hu, Z. Lin, J. Feng, and J. Zhou. Smooth representation clustering. In *CVPR*, 2014. 2, 4
- [21] K. Huang, Y. Ma, and R. Vidal. Minimum effective dimension for mixtures of subspaces: A robust GPCA algorithm and its applications. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 631–638, 2004. 1
- [22] A. Leonardis, H. Bischof, and J. Maver. Multiple eigenspaces. *Pattern Recognition*, 35(11):2613–2627, 2002. 1
- [23] Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv:1009.5055v2*, 2011. 3, 5
- [24] G. Liu, Z. Lin, S. Yan, J. Sun, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 35(1):171–184, Jan 2013. 1, 3
- [25] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*, 2010. 1, 3, 4, 7, 8
- [26] G. Liu and S. Yan. Latent low-rank representation for subspace segmentation and feature extraction. *International Conference on Computer Vision*, 2011. 2, 8
- [27] C. Lu, Z. Lin, and S. Yan. Correlation adaptive subspace segmentation by trace lasso. In *ICCV*, 2013. 1, 2, 3, 4, 8
- [28] C. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan. Robust and efficient subspace segmentation via least squares regression. In *Proceedings of European Conference on Computer Vision*, 2012. 1, 2, 3, 4, 7, 8
- [29] C. Lu, S. Yan, and Z. Lin. Correntropy induced l2 graph for robust subspace clustering. In *ICCV*, 2013. 1, 2
- [30] D. Luo, F. Nie, C. H. Q. Ding, and H. Huang. Multi-subspace representation and discovery. In *ECML/PKDD*, pages 405–420, 2011. 2
- [31] Y. Ma, H. Derksen, W. Hong, and J. Wright. Segmentation of multivariate mixed data via lossy coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1546–1562, 2007. 1
- [32] Y. Ma, A. Y. Yang, H. Derksen, and R. Fossum. Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM Review*, 50(3):413–458, 2008. 1
- [33] V. M. Patel, H. V. Nguyen, and R. Vidal. Latent space sparse subspace clustering. In *IEEE International Conference on Computer Vision*, 2013. 2
- [34] V. M. Patel and R. Vidal. Kernel sparse subspace clustering. In *International Conference on Image Processing*, 2014. 2
- [35] D. Pham, S. Budhacharya, D. Phung, and S. Venkatesh. Improved subspace clustering via exploitation of spatial constraints. In *CVPR*, 2012. 2, 4

- [36] S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1832–1845, 2010. [1](#)
- [37] M. Soltanolkotabi and E. J. Candès. A geometric analysis of subspace clustering with outliers. *Annals of Statistics*, 40(4):2195–2238, 2012. [3](#), [4](#)
- [38] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography. *International Journal of Computer Vision*, 9(2):137–154, 1992. [1](#), [7](#)
- [39] R. Tron and R. Vidal. A benchmark for the comparison of 3-D motion segmentation algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007. [6](#), [7](#)
- [40] P. Tseng. Nearest q -flat to m points. *Journal of Optimization Theory and Applications*, 105(1):249–252, 2000. [1](#)
- [41] R. Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(3):52–68, March 2011. [1](#)
- [42] R. Vidal and P. Favaro. Low rank subspace clustering (LRSC). *Pattern Recognition Letters*, 43:47–61, 2014. [1](#), [8](#)
- [43] R. Vidal, Y. Ma, and S. Sastry. Generalized Principal Component Analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1–15, 2005. [1](#)
- [44] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17, 2007. [6](#)
- [45] S. Wang, X. Yuan, T. Yao, S. Yan, and J. Shen. Efficient subspace segmentation via quadratic programming. In *AAAI*, 2011. [2](#)
- [46] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *European Conference on Computer Vision*, pages 94–106, 2006. [1](#), [7](#)
- [47] A. Y. Yang, S. Rao, and Y. Ma. Robust statistical estimation and segmentation of multiple subspaces. In *Workshop on 25 years of RANSAC*, 2006. [1](#)
- [48] T. Zhang, A. Szlam, and G. Lerman. Median k -flats for hybrid linear modeling with many outliers. In *Workshop on Subspace Methods*, 2009. [1](#)
- [49] T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Hybrid linear modeling via local best-fit flats. *International Journal of Computer Vision*, 100(3):217–240, 2012. [1](#)