

Nearness to Local Subspace Algorithm for Subspace and Motion Segmentation

Akram Aldroubi and Ali Sekmen, *Member, IEEE*

Abstract—This letter presents a clustering algorithm for high dimensional data that comes from a union of lower dimensional subspaces of equal and known dimensions. The algorithm estimates a local subspace for each data point, and computes the distances between the local subspaces and the points to convert the problem to a one-dimensional data clustering problem. The algorithm is reliable in the presence of noise, and applied to the Hopkins 155 Dataset, it generates the best results to date for motion segmentation. The two motion, three motion, and overall segmentation rates for the video sequences are 99.43%, 98.69%, and 99.24%, respectively.

Index Terms—Similarity matrix, spectral clustering, subspace segmentation, unions of subspaces.

I. INTRODUCTION

IN many engineering and mathematics applications, data lives in a union of low dimensional subspaces [1]–[4]. For instance, the set of all two dimensional images of a given face i , obtained under different illuminations and facial positions, can be modeled as a set of vectors that belongs to a low dimensional subspace S_i living in a higher dimensional space \mathbb{R}^D [5], [6]. A set of such images from different faces is then a union $\mathcal{U} = \bigcup_{i \in I} S_i$, where I is an indexing set. Similar nonlinear models arise in sampling theory where \mathbb{R}^D is replaced by an infinite dimensional Hilbert space \mathcal{H} , e.g., $L^2(\mathbb{R}^D)$ [1], [7]. The problem of subspace clustering is to find a model of the form $\mathcal{U} = \bigcup_{i \in I} S_i$ where $\{S_i\}_{i \in I}$ is a set of subspaces that is nearest to a set of data $\mathbf{W} = \{w_1, \dots, w_N\} \in \mathbb{R}^D$. The model can then be used to classify the data \mathbf{W} into classes called clusters. The number of subspaces, their dimensions, and a basis for each subspace are to be determined even in presence of noise, missing data, and outliers. A number of approaches have been devised to solve the problem above or some of its special cases. They are based on sparsity methods [8]–[10], algebraic methods [11], [12], iterative and statistical methods [2], [3], [13], [14], and spectral clustering methods [15]–[18].

Motion segmentation is a special case of the general subspace segmentation problem [19]. Consider a moving affine camera that captures F frames of a scene that contains multiple moving

rigid objects. It can be shown that the trajectory vectors of all points of an object in a video belong to a vector subspace in \mathbb{R}^{2F} of dimension no larger than four [20]. Thus, trajectory vectors in videos can be modeled by a union $\mathcal{M} = \bigcup_{i \in I} V_i$ of l subspaces, where l is the number of moving objects.

A. Letter Contributions

This letter presents a clustering algorithm for high dimensional data that are drawn from a union of low dimensional subspaces of equal and known dimensions. The algorithm is applicable to the motion segmentation problem and uses some fundamental linear algebra concepts. Some of our ideas are similar to those of Yan and Pollefeys [18]. However, our algorithm differs from theirs fundamentally. Yan and Pollefeys' method estimate a subspace S_i for each vector (i.e., trajectory vector) x_i , and then computes the principle angles between those subspaces as an affinity measure. In our work, we also estimate a subspace for each point, however, these local subspaces are used differently. They are used to compute the distance between each point x_j to the local subspace S_i for the data point x_i . In their method, an exponential function for affinity of two points x_i and x_j is used, and this exponential function depends on the principle angles between the subspaces S_i and S_j that are associated with x_i and x_j , respectively. In our case, the affinity measure is different. We first find the distance between x_j and S_i and then apply a threshold, computed from the data, to obtain a binary similarity matrix for all data points. The method of Yan and Pollefeys uses spectral clustering on the normalized graph Laplacian matrix of the similarity matrix they propose. However, our approach does not use the spectral clustering on the normalized graph Laplacian of our similarity matrix. Instead, our constructed binary similarity matrix converts our original data clustering problem to a simpler clustering of data drawn from 1-dimensional subspaces.

Our algorithm is reliable in the presence of noise, and applied to the Hopkins 155 Dataset, it generates the best results to date for motion segmentation.

Many of the subspace segmentation algorithms use SVD to represent the data matrix W as $W = U\Sigma V^t$ and then replace W with the first r rows of V^t , where r is the effective rank of W . This letter provides a formal justification for this in Proposition 1.

II. NEARNESS TO LOCAL SUBSPACE APPROACH

In this section, we develop a specialized algorithm for subspace segmentation and data clustering when the dimensions of the subspaces are equal and known. First, a local subspace is estimated for each data point. Then, the distances between the local subspaces and points are computed and a distance matrix is generated. This is followed by construction of a binary similarity matrix by applying a data-driven threshold to the distance matrix. Finally, the segmentation problem is converted to a one-dimensional data clustering problem.

Manuscript received July 03, 2012; revised August 03, 2012; accepted August 13, 2012. Date of publication August 20, 2012; date of current version August 28, 2012. The work of A. Aldroubi was supported in part by NSF Grant DMS-1108631. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Gustavo Kunde Rohde.

A. Aldroubi is with the Department Mathematics, Vanderbilt University, Nashville, TN, 37212 USA (e-mail: akram.alldroubi@vanderbilt.edu).

A. Sekmen is with the Department Mathematics, Vanderbilt University, Nashville, TN, 37212 USA, and also with the Department of Computer Science at Tennessee State University, Nashville, TN 37209 USA (e-mail: asekmn@tnstate.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/LSP.2012.2214211

A. Algorithm for Subspace Segmentation for Subspaces of Equal and Known Dimensions

The algorithm for subspace segmentation is given in Algorithm 1. We assume that the subspaces have dimension d (for rigid motion segmentation, $d \leq 4$). The details of the various steps are as follows.

Algorithm 1 Subspace Segmentation

Require: The $m \times N$ data matrix W whose columns are drawn from subspaces of dimension d

Ensure: Clustering of the feature points.

- 1: Compute the SVD of W as in (II.1).
 - 2: Estimate the rank of W (denoted by r) if it is not known.
 - 3: Compute $(V_r)^t$ consisting of the first r rows of V^t .
 - 4: Normalize the columns of $(V_r)^t$.
 - 5: Replace the data matrix W with $(V_r)^t$.
 - 6: Find the angle between the column vectors of W and represent it as a matrix. {i.e., $\arccos(W^t W)$.}
 - 7: Sort the angles and find the closest neighbors of column vector.
 - 8: **for all** Column vector x_i of W **do**
 - 9: Find the local subspace for the set consisting of x_i and k neighbors (see (II.2)). {Theoretically, k is at least $d - 1$. We can use the least square approximation for the subspace (see the section *Local Subspace Estimation*). Let A_i denote the matrix whose columns form an orthonormal bases for the local subspace associated with x_i .}
 - 10: **end for**
 - 11: **for** $i = 1$ to N **do**
 - 12: **for** $j = 1$ to N **do**
 - 13: define $H = (d_{ij}) = (\|x_j - A_i A_i^t x_j\|_p + \|x_i - A_j A_j^t x_i\|_p) / 2$
 - 14: **end for**
 - 15: **end for** {Build the distance matrix}
 - 16: Sort the entries of the $N \times N$ matrix H from smallest to highest values into the vector h and set the threshold η to the value of the T^{th} entry of the sorted and normalized vector h , where T is such that $\|\chi_{[T, N^2]} - h\|_2$ is minimized, and where $\chi_{[T, N^2]}$ is the characteristic function of the discrete set $[T, N^2]$.
 - 17: Construct a similarity matrix S by setting all entries of H less than threshold η to 1 and by setting all other entries to 0. {Build the binary similarity matrix}
 - 18: Normalize the rows of S using l_1 -norm.
 - 19: Perform SVD $S^t = U_n \Sigma_n (V_n)^t$.
 - 20: Cluster the columns of $\Sigma_n (V_n)^t$ using k-means. $\Sigma_n (V_n)^t$ is the projection on to the span of U_n .
-

Dimensionality Reduction and Normalization: Let W be an $m \times N$ data matrix whose columns are drawn from a union

of subspaces of dimensions at most d , possibly perturbed by noise. In order to reduce the dimensionality of the problem, we compute the SVD of W

$$W = U \Sigma V^t \quad (\text{II.1})$$

where $U = [u_1 \ u_2 \ \dots \ u_m]$ is an $m \times m$ matrix, $V = [v_1 \ v_2 \ \dots \ v_N]$ is an $N \times N$ matrix, and Σ is an $m \times N$ diagonal matrix with diagonal entries $\sigma_1, \dots, \sigma_l$, where $l = \min\{m, N\}$. To estimate the effective rank of W , one can use the modal selection algorithm [18], [21] to estimate the rank r if it is not known. We can now replace the data matrix W with the matrix $(V_r)^t$ that consists of the first r rows of V^t (thereby reducing the dimensionality of data). This step is justified by the following proposition which is used to validate that a data matrix W whose columns represent data points can be replaced with a lower rank matrix after computing its SVD (i.e. $W = U \Sigma V^t$). It can be paraphrased by saying that for any matrices A, B, C , a cluster of the columns of B is also a cluster of the columns of $C = AB$. A cluster of C however is not necessarily a cluster B , unless A has full rank:

Proposition 1: Let A and B be $m \times n$ and $n \times k$ matrices. Let $C = AB$. Assume $J \subset \{1, 2, \dots, k\}$.

- 1) If $b_i \in \text{span}\{b_j : j \in J\}$ then $c_i \in \text{span}\{c_j : j \in J\}$.
- 2) If A is full row rank (thus, $m \geq n$) then $b_i \in \text{span}\{b_j : j \in J\} \iff c_i \in \text{span}\{c_j : j \in J\}$

Proof: It is a straightforward application of linear algebra. ■

Also, [11] discusses the segmentation preserving projections and states that the number of subspaces and their dimensions are preserved by random projections, except for a zero measure set of projections. It should also be noted that this step reduces additive noise as well, especially in the case of light-tailed noise, e.g., Gaussian noise. Dimensionality reduction corresponds to Steps 1, 2, and 3 in Algorithm 1.

Another type of data reduction is normalization. Specifically, the columns of $(V_r)^t$ are normalized to lie on the unit sphere \mathbb{S}^{r-1} . This is because by projecting the subspace on the unit sphere, we effectively reduce the dimensionality of the data by one. Moreover, the normalization gives equal contribution of the data matrix columns to the description of the subspaces. Note that the normalization can be done by using l_p norms of the columns of $(V_r)^t$. This normalization procedure corresponds to Steps 4 and 5 in Algorithm 1.

Local Subspace Estimation: The trajectory vectors that are close to each other are likely to belong to the same subspace. For this reason, we estimate a local subspace for each data point using its closest neighbors. This can be done in different ways. For example, if the l_2 -norm is used for normalization, we can find the angles between the trajectories, i.e., we can compute the matrix $\arccos(V_r \times (V_r)^t)$. Then we can sort the angles and find the closest neighbors of each point. If we use l_p -norm for normalization, we can generate a distance matrix $(a_{ij}) = (\|x_i - x_j\|_p)$ and then sort each column of the distance matrix to find the neighbors of each x_i , which is the i^{th} column of $(V_r)^t$. Once the distance matrix between the points is generated, we can find, for each point x_i , a set of $k + 1 \geq d$ points $\{x_i, x_{i_1}, \dots, x_{i_k}\}$ consisting of x_i and its k closest neighbors. Then we generate a d -dimensional subspace that is nearest (in the least square sense) to the data $\{x_i, x_{i_1}, \dots, x_{i_k}\}$. This is accomplished by using SVD

$$X = [x_i \ x_{i_1} \ \dots \ x_{i_k}] = A \Sigma B^t \quad (\text{II.2})$$

Let A_i denote the matrix of the first d columns of A associated with x_i . Then, the column space $C(A_i)$ is the d -dimensional subspace nearest to $\{x_i, x_{i_1}, \dots, x_{i_k}\}$. Local subspace estimation corresponds to Steps 6 to 10 in Algorithm 1.

Construction of Binary Similarity Matrix: So far, we have associated a local subspace S_i to each point x_i . Ideally, the points and only those points that belong to the same subspace as x_i should have zero distance from S_i . This suggests computing the distance of each point x_j to the local subspace S_i and forming a distance matrix H .

The distance matrix H is generated as $H = (d_{ij}) = (\|x_j - A_i A_i^t x_j\|_p + \|x_i - A_j A_j^t x_i\|_p) / 2$.

A convenient choice of p is 2. Note that as d_{ij} decreases, the probability of having x_j on the same subspace as x_i increases. Moreover, for $p = 2$, $\|x_j - A_i A_i^t x_j\|_2$ is the Euclidean distance of x_j to the subspace associated with x_i . Since we are not in the ideal case, a point x_j that belongs to the same subspace as x_i may have non-zero distance to S_i . However, this distance is likely to be small compared to the distance between x_j and S_k if x_j and x_k do not belong to the same subspace. This suggests that we compute a threshold that will distinguish between these two cases and transform the distance matrix into a binary matrix in which a zero in the (i, j) entry means x_i and x_j are likely to belong to the same subspace, whereas (i, j) entry of one means x_i and x_j are not likely to belong to the same subspace. To do this, we convert the distance matrix $H = (d_{ij})_{N \times N}$ into a binary similarity matrix $S = (s_{ij})$. This is done by applying a data-driven thresholding as follows:

- 1) Create a vector h that contains the sorted entries of $H_{N \times N}$ from smallest to highest values. Scale and offset h so that its smallest value is zero and its largest value is one.
- 2) Set the threshold η to the value of the T^{th} entry of the sorted vector h , where T is such that $\|\chi_{[T, N^2]} - h\|_2$ is minimized, and where $\chi_{[T, N^2]}$ is the characteristic function of the discrete set $[T, N^2]$. If the number of points in each subspace are approximately equal, then we would expect about N/n points in each subspace, and we would expect N^2/n^2 small entries (zero entries ideally). However, this may not be the case in general. For this reason, we compute the data-driven threshold η that distinguishes the small entries from the large entries.
- 3) Create a similarity matrix S from H such that all entries of H less than the threshold η are set to 1 and the others are set to 0.

The construction of binary similarity corresponds to Steps 11 to 17 in Algorithm 1. In [18], Yan and Pollofeys uses chordal distance between the subspaces $\mathcal{F}(x_i)$ and $\mathcal{G}(x_j)$ as a measure of the distance between points x_i and x_j

$$d_c^2(\mathcal{F}, \mathcal{G}) = \sum_{i=1}^p \sin^2(\theta_i) \quad (\text{II.3})$$

where $\{\theta_i\}_{i=1}^p$ are the principle angles between p -dimensional local subspaces \mathcal{F} and \mathcal{G} with $\theta_1 \leq \dots \leq \theta_p$. In this approach, the distance between any pairs of points from \mathcal{F} and \mathcal{G} is the same. We find distances between points and local subspaces and our approach distinguishes different points from the same subspace. To see this, let $v \in \text{span}\{Q_{\mathcal{F}}\}$, $\|v\|_2 = 1$, where the columns of $Q_{\mathcal{F}}$ form an orthonormal basis for \mathcal{F} . Thus $v = Q_{\mathcal{F}}x$ for some x with $\|x\|_2 = 1$. Let $Q_{\mathcal{G}}$ form an

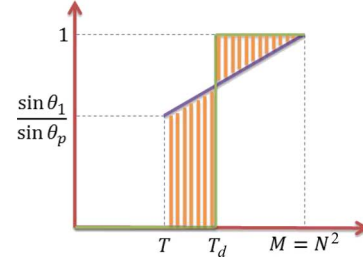


Fig. 1. Linear modeling for h in the ideal noiseless case.

orthonormal basis for \mathcal{G} , then the Euclidian distance from v to \mathcal{G} squared is given by

$$\|v - P_{\mathcal{G}}(v)\|_2^2 = \|Q_{\mathcal{F}}x - Q_{\mathcal{G}}Q_{\mathcal{G}}^t Q_{\mathcal{F}}x\|_2^2 = z^t(I - \Sigma^2)z$$

where $Y\Sigma Z^t$ is the SVD for $Q_{\mathcal{F}}^t Q_{\mathcal{G}}$ and $z := Y^t x$. Thus, using the relation $\cos \theta_i = \sigma_i$ between principle angles and singular values, we get

$$d^2(v, \mathcal{G}) = \sum_{i=1}^p z_i^2 \sin^2(\theta_i). \quad (\text{II.4})$$

Hence, our approach discriminates distances from points in \mathcal{F} to subspace \mathcal{G} . We also have $\sum_{i=1}^p z_i^2 \sin^2(\theta_i) \leq \sum_{i=1}^p \sin^2(\theta_i)$. Using (II.4), we get $0 < \sin \theta_1 \leq d \leq \sin \theta_p$. Assuming a uniform distribution of samples from \mathcal{F} and \mathcal{G} , h can be approximated by a linear function depicted in Fig. 1. This is only an approximation that depends on θ_1 and θ_p . The exact function will not be linear and will depend on all the angles. The goal is to find the threshold at the jump discontinuity T from 0 to $\sin \theta_1$. Our method minimizes the highlighted area. Under this model, a simple computation shows that our data driven thresholding algorithm selects the correct threshold if $\sin \theta_1 / \sin \theta_p \geq 1/2$, e.g., if $\theta_1 \geq 30^\circ$. In other situations, our algorithm overshoots in estimating the threshold index depending on θ_1 and θ_p .

Segmentation: The last step is to use the similarity matrix S to segment the data. To do this, we first normalize the rows of S using l_1 -norm, i.e., $\tilde{S} = D^{-1}S$, where D is a diagonal matrix $(d_{ij}) = \sum_{j=1}^N s_{ij}$. \tilde{S} is related to the random walk Laplacian L_r ($\tilde{S} = I - L_r$). Other l_p normalizations are also possible for $p \geq 1$, however, l_1 -normalization brings outliers closer to the cluster clouds (distances of outliers decrease monotonically as p decreases to 1). This is due to the geometry of the l_1 ball. Since SVD (which will be used next) is associated with l_2 minimization it is sensitive to outliers. Therefore l_1 normalization works best when SVD is used. Observe that the initial data segmentation problem has now been converted to segmentation of n 1-dimensional subspaces from the rows of \tilde{S} . This is because, in the ideal case, from the construction of \tilde{S} , if x_i and x_j are in the same subspace, the i^{th} and j^{th} rows of \tilde{S} are equal. Since there are n subspaces, then there will be n 1-dimensional subspaces. Now, the problem is again a subspace segmentation problem, but this time the data matrix is \tilde{S} with each row as a data point. Also, each subspace is 1-dimensional and there are n subspaces. Therefore, we can apply SVD again to obtain $\tilde{S}^t = U_n \Sigma_n (V_n)^t$. Using Proposition 1, it can be shown that $\Sigma_n (V_n)^t$ can replace \tilde{S}^t and we cluster the columns of $\Sigma_n (V_n)^t$, which is the projection of \tilde{S} on to the span of U_n . Since the problem is only

TABLE I
% SEGMENTATION ERRORS FOR SEQUENCES WITH TWO AND THREE MOTIONS

Average	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS
Checker-2 (78)	6.09%	2.57%	6.52%	4.46%	1.55%	0.83%	1.12%	0.23%
Traffic-2 (31)	1.41%	5.43%	2.55%	2.23%	1.59%	0.23%	0.02%	1.40%
Articulated-2 (11)	2.88%	4.10%	7.25%	7.23%	10.70%	1.63%	0.62%	1.77%
All-2 (120 seq)	4.59%	3.45%	5.56%	4.14%	2.40%	0.75%	0.82%	0.57%
Checker-3 (26)	31.95%	5.80%	25.78%	10.38%	5.20%	4.49%	2.97%	0.87%
Traffic-3 (7)	19.83%	25.07%	12.83%	1.80%	7.75%	0.61%	0.58%	1.86%
Articulated-3 (2)	16.85%	7.25%	21.38%	2.71%	21.08%	1.60%	1.60%	5.12%
All-3 (35 seq)	28.66%	9.73%	22.94%	8.23%	6.69%	3.55%	2.45%	1.31%
All (155 seq)	10.34%	4.94%	9.76%	5.03%	3.56%	1.45%	1.24%	0.76%

TABLE II
% COMPARISON OF THE DATA DRIVEN THRESHOLD INDEX T_d WITH OTHER CHOICES

Average	Data Driven T_d	$0.8T_d$	$0.9T_d$	$0.95T_d$	$1.05T_d$	$1.10T_d$	$1.20T_d$
All-2 (120 seq)	0.57%	0.95%	1.17%	0.62%	0.58%	1.05%	0.77%
All-3 (35 seq)	1.31%	4.39%	3.18%	1.42%	1.20%	1.24%	2.06%
All (155 seq)	0.76%	1.84%	1.67%	0.83%	0.74%	1.10%	1.11%

TABLE III
% SEGMENTATION ERRORS—NLS ALGORITHM FOR VARIOUS k

Average	k=5	k=4	k=3
All-2 (120 seq)	1.04%	1.75%	0.57%
All-3 (35 seq)	2.82%	1.98%	1.31%
All (155 seq)	1.50%	1.81%	0.76%

TABLE IV
% SEGMENTATION ERRORS FOR LSA WITH VARIOUS PARAMETERS

Average	LSA (Original)	LSA (New Dist/Similarity)	LSA (New Segmentation)
All-2 (120 seq)	3.45%	1.22%	2.27%
All-3 (35 seq)	9.73%	2.45%	8.78%
All (155 seq)	4.94%	1.84%	3.96%

segmentation of subspaces of dimension 1, we can use any traditional segmentation algorithm such as k-means to cluster the data points. The segmentation corresponds to Steps 18 to 20 in Algorithm 1.

III. EXPERIMENTAL RESULTS

Table I displays some of the experimental results for the Hopkins 155 Dataset [12]. Our Nearness to Local Subspace (NLS) approach have been compared with: (1) GPCA, (2) RANSAC, (3) Local Subspace Affinity (LSA), (4) MLS, (5) Agglomerative Lossy Compression, and (6) Sparse Subspace Clustering (SSC). An evaluation of those algorithms is presented in [15] with a minor error (the listing of error as 1.42% for articulated three motions instead of 1.60%). SSC-B and SSC-N correspond to Bernoulli and Normal random projections [15]. Table I used the number of neighbors $k = 3$. Since each point is drawn from a 4-dimensional subspace, a minimum of 3 neighbors are needed to fit a local subspace for each point. Using the same assumption as the algorithms that we compare with, we take the rank of the data matrix to be 8 for two motions and 12 for three motions. NLS has 0.76% misclassification rate compared to 1.24% of the next best algorithm. Table II shows the performance of the data driven threshold index T_d compared to various other possible thresholds. Table III displays the robustness of the algorithm with respect to the number of neighbors k . Table IV displays the increase in the performance of the original LSA algorithm when our distance/similarity and segmentation techniques are applied separately. Both of them improves the performance of the algorithm, however, the new distance and similarity combination contributes more than the new segmentation technique. Recently, the Low-Rank Representation (LRR) in [9] was ap-

plied to the Hopkins 155 Datasets and it generated an error rate of 3.16%. The authors state that this error rate can be reduced to 0.87% by using a variation of LRR with some additional adjustment of a certain parameter.

IV. CONCLUSIONS

The NLS approach described in this letter works only in the cases of subspaces of equal and known dimensions. It is based on the computation of a binary similarity matrix for the data points. A local subspace is first estimated for each data point. Then, a distance matrix is generated by computing the distances between the local subspaces and points. The distance matrix is converted to the similarity matrix by applying a data-driven threshold. The problem is then transformed to segmentation of subspaces of dimension 1. The algorithm was applied to the Hopkins 155 Dataset and generated the best results to date.

REFERENCES

- [1] Y. M. Lu and M. N. Do, "A theory for sampling signals from a union of subspaces," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2334–2345, 2008.
- [2] K. Kanatani and Y. Sugaya, "Multi-stage optimization for multi-body motion segmentation," *IEICE Trans. Inf. Syst.*, pp. 335–349, 2003.
- [3] A. Aldroubi and K. Zaringhalam, "Nonlinear least squares in \mathbb{R}^n ," *Acta Appl. Math.*, vol. 107, no. 1–3, pp. 325–337, 2009.
- [4] G. Chen, A. V. Little, M. Maggioni, and L. Rosasco, "Some recent advances in multiscale geometric analysis of point clouds," in *Wavelets and Multiscale Analysis, Applied and Numerical Harmonic Analysis*, J. Cohen and A. I. Zayed, Eds. Boston, MA: Birkhuser, 2011, pp. 199–225.
- [5] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, pp. 218–233, 2003, 2.
- [6] J. Ho, M. Yang, J. Lim, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2003, pp. 11–18.
- [7] A. Aldroubi and R. Tessera, "On the existence of optimal unions of subspaces for data modeling and clustering," *Found. Comput. Math.*, 2011, p. arXiv:1008.4811v1, to appear.
- [8] E. Elhamifar and R. Vidal, "Clustering disjoint subspaces via sparse representation," in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2010.
- [9] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Int. Conf. Machine Learning*, 2010, pp. 663–670.
- [10] P. Favaro, R. Vidal, and A. Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2011.
- [11] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1945–1959, 2005, 12.
- [12] R. Tron and R. Vidal, "A benchmark for the comparison of 3-d motion segmentation algorithms," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
- [13] P. Tseng, "Nearest q-flat to m points," *J. Optim. Theory Applicat.*, vol. 105, no. 1, pp. 249–252, 2000.
- [14] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981, 6.
- [15] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2009, pp. 2790–2797.
- [16] U. V. Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, pp. 395–416, 2007.
- [17] G. Chen and G. Lerman, "Spectral curvature clustering (SCC)," *Int. J. Comput. Vis.*, vol. 81, pp. 317–330, 2009.
- [18] J. Yan and M. Pollefeys, "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and nondegenerate," in *9th Eur. Conf. Computer Vision*, 2006, pp. 94–106.
- [19] R. Vidal, "A tutorial on subspace clustering," *IEEE Signal Process. Mag.*, 2010.
- [20] K. Kanatani, "Motion segmentation by subspace separation and model selection," in *8th Int. Conf. Computer Vision*, 2001, vol. 2, pp. 301–306, 7.
- [21] L. Zappella, X. Lladó, E. Provenzi, and J. Salvi, "Enhanced local subspace affinity for feature-based motion segmentation," *Pattern Recognit.*, vol. 44, pp. 454–470, 2011.