Latent Space Sparse and Low-rank Subspace Clustering

Vishal M. Patel, *Member, IEEE,* Hien V. Nguyen, *Student Member, IEEE,* and René Vidal, *Senior Member, IEEE*

***Abstract*—We propose three novel algorithms for simultaneous dimensionality reduction and clustering of data lying in a union of subspaces. Specifically, we describe methods that learn the projection of data and find the sparse and/or low-rank coefficients in the low-dimensional latent space. Cluster labels are then assigned by applying spectral clustering to a similarity matrix built from these representations. Efficient optimization methods are proposed and their non-linear extensions based on the kernel methods are presented. Various experiments show that the proposed methods perform better than many competitive subspace clustering methods.**

***Index Terms*—Subspace clustering, sparse subspace clustering, low-rank subspace clustering, kernel methods, non-linear subspace clustering, dimension reduction.**

I. Introduction

Many practical computer vision and image processing applications require processing and representation of high-dimensional data. Often these high-dimensional data can be represented by a low-dimensional subspace. For instance, it is well known that the set of face images under all possible illumination conditions can be well approximated by a 9-dimensional linear subspace [1]. Similarly, trajectories of a rigidly moving object in a video [2] and hand-written digits with different variations [3] also lie in low-dimensional subspaces. Therefore, one can view the collection of data from different classes as samples from a union of low-dimensional subspaces. In subspace clustering, given the data from a union of subspaces, the objective is to find the number of subspaces, their dimensions, the segmentation of the data and a basis for each subspace (formal definition is given in Section II) [4].

Various algorithms have been proposed in the literature for subspace clustering. Some of these algorithms are iterative in nature [5], [6], [7] while the others are based on spectral clustering [8], [9], [10], [11]. Statistical [12] and algebraic [13], [14] approaches have also been proposed in the literature for subspace clustering. In particular, sparse representation and low-rank approximation-based methods for subspace clustering [15], [16], [11], [17], [18], [19], [20], [21], [22], [23], [24] have gained a lot of traction in recent years. These methods find a sparse or low-rank representation of the data and build a similarity graph whose weights depend on the sparse or low-rank coefficient matrix for segmenting the data. One of the advantages of these methods is that they are robust to noise and occlusion. Furthermore, some of these approaches do not require the knowledge of the dimensions and the number of subspaces. In particular, the Sparse Subspace Clustering (SSC) algorithm [11], [17], Low-Rank Representation (LRR) based algorithm [15] and Low-Rank Sparse Subspace Clustering (LRSSC) method [23] are well supported by theoretical analysis [25] [18], [19], [23] and provide state-of-the-art results on many publicly available datasets such as the Hopkins155 benchmark motion segmentation dataset [26].

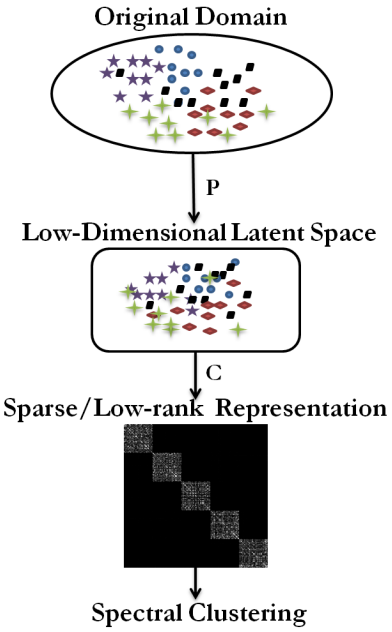


Fig. 1: Overview of the proposed latent space sparse and low-rank subspace clustering methods.

Finding sparse or low-rank representation is very computationally demanding especially when the dimension of the features is high [17]. This is one of the drawbacks of the sparse and low-rank representation-based methods. To deal with this problem, dimensionality reduction is generally applied on the data prior to applying these algorithms. Dimensionality reduction methods such as Principle Component Analysis (PCA) and Random Projections (RP) can reduce the dimension of data. However, a well learned projection matrix can lead to

Vishal M. Patel is with the Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742 USA (e-mail: pvishalm@umd.edu).

Hien V. Nguyen is with the Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742 USA (e-mail: hien@umd.edu).

René Vidal is with the the Center for Imaging Science, Department of Biomedical Engineering, The Johns Hopkins University, 302B Clark Hall, 3400 N. Charles St., Baltimore MD 21218 USA (e-mail: rvidal@cis.jhu.edu).

a higher clustering accuracy at a lower dimensionality. Several works have been proposed in the literature that find a sparse representation on a low-dimensional latent space [27], [28], [29]. However, these methods are specifically designed for classification tasks and not for clustering.

Motivated by some of the sparsity promoting dimensionality reduction methods, in this paper, we propose methods for simultaneous dimensionality reduction and subspace clustering under the frameworks of SSC, LRR and LRSSC. We learn the transformation of data from the original space onto a low-dimensional space such that its manifold structure is maintained. Efficient algorithms are proposed that simultaneously learn the projection and find the sparse or low-rank coefficients in the low-dimensional latent space. Finally, the segmentation of the data is obtained by applying spectral clustering to a similarity matrix built from these representation coefficients. Using kernel methods, the proposed algorithms are also extended to non-linear manifolds. Figure 1 presents an overview of our latent space subspace clustering methods.

Key contributions of our work are as follows:

- Simultaneous dimensionality reduction and low-rank and/or sparse representation methods for subspace clustering are proposed.
- Simple iterative procedures are introduced for solving the proposed optimization problems.
- Nonlinear extensions of the proposed algorithms are made through the use of Mercer kernels.

This paper is organized as follows. In Section II, we provide a brief overview of the sparse and low-rank subspace clustering methods. Sections III and IV give the details of our linear and non-linear simultaneous dimensionality reduction and subspace clustering approaches, respectively. Experimental results are presented in Section V and Section VI concludes the paper with a brief summary and discussion.

II. Background

In this section, we provide a brief background on sparse and low-rank subspace clustering methods [17], [18], [19], [23].

A. Problem Formulation

Let

$$\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N] \in \mathbb{R}^{D \times N}$$

be a collection of N signals $\{\mathbf{y}_i \in \mathbb{R}^D\}_{i=1}^N$ drawn from a union of n linear subspaces

$$\mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_n$$

of dimensions $\{d_\ell\}_{\ell=1}^n$ in $\mathbb{R}^D$. Let $\mathbf{Y}_\ell \in \mathbb{R}^{D \times N_\ell}$ be a submatrix of $\mathbf{Y}$ of rank d_ℓ with $N_\ell > d_\ell$ points that lie in $\mathcal{S}_\ell$ with $N_1 + N_2 + \cdots + N_n = N$. Given $\mathbf{Y}$, the task of subspace clustering is to cluster the signals according to their subspaces.

B. Sparse Subspace Clustering

It is easy to see that each data point in $\mathbf{Y}$ can be efficiently represented by a linear combination of at most d_ℓ other points in $\mathbf{Y}$. That is, one can represent $\mathbf{y}_i$ as follows

$$\mathbf{y}_i = \mathbf{Y}\mathbf{c}_i, \;\; c_{ii} = 0, \; \|\mathbf{c}_i\|_0 \leq d_\ell$$

where $\mathbf{c}_i = [c_{i1}, c_{i2}, \cdots, c_{iN}]^T \in \mathbb{R}^N$ are the coefficients and $\|\mathbf{x}\|_0$ is the sparsity measure that counts the number of non-zero elements in $\mathbf{x}$. Often $N_\ell > d_\ell$. As a result the following ℓ_1-minimization problem is solved to obtain the coefficients

$$\min \|\mathbf{c}\|_1 \text{ such that } \mathbf{y}_i = \mathbf{Y}\mathbf{c}_i, \; c_{ii} = 0, \tag{1}$$

where $\|x\|_1 = \sum_{i=1}^N |x_i|$ is the ℓ_1-norm of $\mathbf{x} \in \mathbb{R}^N$. Considering all the data points $i = 1, \cdots, N$, in matrix form, the above optimization problem can be rewritten as

$$\min \|\mathbf{C}\|_1 \;\text{ subject to }\; \mathbf{Y} = \mathbf{Y}\mathbf{C}, \;\; \text{diag}(\mathbf{C}) = \mathbf{0}, \tag{2}$$

where $\mathbf{C} = [\mathbf{c}_1, \cdots, \mathbf{c}_N] \in \mathbb{R}^{N \times N}$ is the coefficient matrix whose column $\mathbf{c}_i$ is the sparse representation vector corresponding to $\mathbf{y}_i$, $\text{diag}(\mathbf{C}) \in \mathbb{R}^N$ is the vector containing the diagonal elements of $\mathbf{C}$ and $\mathbf{0} \in \mathbb{R}^N$ is an N-dimensional vector containing zeros as its elements.

In some applications, the data lie in a union of affine rather than linear subspaces. To deal with affine subspaces, we use the fact that any point $\mathbf{y}_i$ in an affine subspace $\mathcal{S}_\ell$ of dimension d_ℓ can be written as an affine combination of $d_\ell + 1$ other points from $\mathcal{S}_\ell$ [17]. In other words, one can represent $\mathbf{y}_i$ as follows

$$\mathbf{y}_i = \mathbf{Y}\mathbf{c}_i, \;\; \mathbf{c}_i^T\mathbf{1} = 1, \;\; c_{ii} = 0,$$

where $\mathbf{1}$ is a vector of dimension N containing ones as its elements. In the case where the data is contaminated by some arbitrary noise $\mathbf{Z}$, i.e. $\mathbf{Y} = \mathbf{Y}\mathbf{C} + \mathbf{Z}$, and considering the fact that data may lie in a union of affine subspaces, the following problem can be solved to obtain $\mathbf{C}$

$$\begin{aligned} \min \|\mathbf{C}\|_1 + \frac{\tau}{2}\|\mathbf{Y} - \mathbf{Y}\mathbf{C}\|_F^2, \\ \text{such that } \; \text{diag}(\mathbf{C}) = \mathbf{0}, \;\; \mathbf{C}^T\mathbf{1} = \mathbf{1}, \end{aligned} \tag{3}$$

where τ is a regularization parameter. The above problems can be efficiently solved by using the classical alternating direction method of multipliers (ADMM) [30], [17].

C. Low-Rank Representation-based Subspace Clustering

The LRR algorithm for subspace clustering is very similar to the SSC algorithm except that a low-rank representation is found instead of a sparse representation. This makes sense because in the case of n independent subspaces of dimensions $\tau = \{d_\ell\}_{\ell=1}^n$, the rank of the data matrix $\mathbf{Y}$ is $\sum_{\ell=1}^n d_\ell$. A collection of subspaces $\{\mathcal{S}_\ell\}_{\ell=1}^n$ is independent if $\dim(\bigoplus_{\ell=1}^n \mathcal{S}_\ell) = \sum_{\ell=1}^n \dim(\mathcal{S}_\ell)$, where $\bigoplus$ denotes the direct sum operator. In the case when the data is noise free, the following rank minimization problem is considered

$$\min_{\mathbf{C}} \text{rank}(\mathbf{C}) \quad \text{such that} \quad \mathbf{Y} = \mathbf{Y}\mathbf{C}. \tag{4}$$

As a common practice in rank minimization problems, the rank of $\mathbf{C}$ is replaced by its nuclear norm $\|\mathbf{C}\|_*$ which is defined as the sum of its singular values. As a result, the following convex problem is solved

$$\min_{\mathbf{C}} \|\mathbf{C}\|_* \quad \text{such that} \quad \mathbf{Y} = \mathbf{Y}\mathbf{C}. \tag{5}$$

It was shown in [18] that the solution to (5) is also a solution to (4). In particular, the following theorem shows that when

$\mathbf{Y}$ is noise free and drawn from n independent subspaces, the optimal solution to (5) can be obtained in closed form [18].

Theorem 1: Suppose the rank r SVD of $\mathbf{Y}$ is $\mathbf{U}_1\mathbf{\Sigma}_1\mathbf{V}_1^T$, then the minimizer to (5) is uniquely given by

$$\hat{\mathbf{C}} = \mathbf{V}_1\mathbf{V}_1^T.$$

In the case, when data is contaminated by noise, the following problem can be solved to approximate $\mathbf{C}$

$$\min_{\mathbf{C}} \|\mathbf{C}\|_* + \frac{\tau}{2}\|\mathbf{Y} - \mathbf{Y}\mathbf{C}\|_F^2. \tag{6}$$

The closed-form solution to this problem has been derived in [16], [19].

Theorem 2: Let $\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ be the SVD of $\mathbf{Y}$ and let σ_i be the ith singular value of $\mathbf{Y}$. The optimal solution to (6) is

$$\hat{\mathbf{C}} = \mathbf{V}_1(\mathbf{I} - \frac{1}{\tau}\mathbf{\Sigma}_1^{-2})\mathbf{V}_1^T,$$

where $\mathbf{U} = [\mathbf{U}_1\mathbf{U}_2], \mathbf{\Sigma} = \text{diag}(\mathbf{\Sigma}_1\mathbf{\Sigma}_2)$, and $\mathbf{V} = [\mathbf{V}_1\mathbf{V}_2]$ are partitioned according to the sets $\mathcal{I}_1 = \{i : \sigma_i > \frac{1}{\sqrt{\tau}}\}$ and $\mathcal{I}_2 = \{i : \sigma_i \leq \frac{1}{\sqrt{\tau}}\}$.

D. Low-rank Sparse Subspace Clustering

The representation matrix $\mathbf{C}$ is often simultaneously sparse and low-rank. As a result, instead of looking for only sparse or low-rank $\mathbf{C}$, one can directly find $\mathbf{C}$ that is both sparse and low-rank. In LRSSC, the following optimization problem is solved to find $\mathbf{C}$

$$\begin{aligned}\min_{\mathbf{C}} \|\mathbf{C}\|_* + \lambda\|\mathbf{C}\|_1 + \frac{\tau}{2}\|\mathbf{Y} - \mathbf{Y}\mathbf{C}\|_F^2 \\ \text{such that } \text{diag}(\mathbf{C}) = \mathbf{0}, \;\; \mathbf{C}^T\mathbf{1} = \mathbf{1}.\end{aligned} \tag{7}$$

This problem can be efficiently solved using the ADMM method [23], [30].

In SSC, LRR and LRSSC, once $\mathbf{C}$ is found, spectral clustering methods [31] are applied on the affinity matrix

$$\mathbf{W} = |\mathbf{C}| + |\mathbf{C}|^T$$

to obtain the segmentation of the data $\mathbf{Y}$ into $\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_n$, where $|\mathbf{C}|$ denotes the modulus of $\mathbf{C}$.

III. Latent Space Subspace Clustering

Different from the traditional sparse and low-rank representation-based subspace clustering methods, we develop algorithms that embed signals into a low-dimensional space and simultaneously find the sparse and/or low-rank representation in that space. Let $\mathbf{P} \in \mathbb{R}^{t\times D}$ be a matrix representing a linear transformation that maps signals from the original space $\mathbb{R}^D$ to a latent output space of dimension t. We can learn the mapping and find the sparse or low-rank representation simultaneously by minimizing the following cost function

$$\begin{aligned}[\mathbf{P}^*, \mathbf{C}^*] = \arg\min_{\mathbf{P},\mathbf{C}} \mathcal{J}_1(\mathbf{C}) + \mathcal{J}_2(\mathbf{P}, \mathbf{C}, \mathbf{Y}) \\ \text{subject to} \quad \mathbf{P}\mathbf{P}^T = \mathbf{I}, \text{diag}(\mathbf{C}) = \mathbf{0},\end{aligned} \tag{8}$$

where

$$\mathcal{J}_2(\mathbf{P}, \mathbf{C}, \mathbf{Y}) = \lambda_1\|\mathbf{P}\mathbf{Y} - \mathbf{P}\mathbf{Y}\mathbf{C}\|_F^2 + \lambda_2\|\mathbf{Y} - \mathbf{P}^T\mathbf{P}\mathbf{Y}\|_F^2 \tag{9}$$

and $\mathcal{J}_1(\mathbf{C})$ depends on whether we find sparse, low-rank or both sparse and low-rank representations. In particular, when sparsity is enforced as is done in SSC, $\mathcal{J}_1(\mathbf{C}) = \|\mathbf{C}\|_1$. Similar to LRR, when low-rank representation is sought, $\mathcal{J}_1(\mathbf{C}) = \|\mathbf{C}\|_*$. Note that in this case, the second constraint $\text{diag}(\mathbf{C}) = \mathbf{0}$ is not required. Finally, one can find both sparse and low-rank representation as is done in LRSSC by setting $\mathcal{J}_1(\mathbf{C}) = \|\mathbf{C}\|_* + \lambda\|\mathbf{C}\|_1$, where λ controls the trade-off between satisfying sparse and low-rank representation.

The second term of $\mathcal{J}_2$ is a PCA-like regularization term, ensures that the projection does not lose too much information available in the original domain. λ_1 and λ_2 are non-negative constants that control sparsity and regularization, respectively. Furthermore, we require the rows of $\mathbf{P}$ to be orthogonal and normalized to unit norm. This prevents the solution from becoming degenerate and the leads to a computationally efficient scheme for optimization. Note that the optimization problem (8) is non-convex. However, numerical simulations have shown that the algorithm usually converges to a local minimum in a few iterations.

The above formulation can be extended so that it can deal with data that lie in a union of affine subspaces. This can be simply done by adding a constraint in the optimization problem (8) as follows

$$\begin{aligned}[\mathbf{P}^*, \mathbf{C}^*] = \arg\min_{\mathbf{P},\mathbf{C}} \mathcal{J}_1(\mathbf{C}) + \mathcal{J}_2(\mathbf{P}, \mathbf{C}, \mathbf{Y}) \\ \text{subject to } \mathbf{P}\mathbf{P}^T = \mathbf{I}, \mathbf{C}^T\mathbf{1} = \mathbf{1}, \text{diag}(\mathbf{C}) = \mathbf{0}.\end{aligned} \tag{10}$$

A. Optimization

With the above definitions, one can prove the following proposition.

Proposition 1: There exists an optimal solution $\mathbf{P}^*$ to (8) that has the following form

$$\mathbf{P}^* = \mathbf{\Psi}^T\mathbf{Y}^T$$

for some $\mathbf{\Psi} \in \mathbb{R}^{N\times t}$.

Intuitively, the above proposition says that the projection can be written as a linear combination of the data samples. This formulation has been used under the framework of dictionary learning in [32] and [33]. The proof of the above proposition can be found in the Appendix.

With this proposition, the cost function $\mathcal{J}_2$ can be written as

$$\begin{aligned}\mathcal{J}_2(\mathbf{\Psi}, \mathbf{C}, \mathbf{Y}) = \lambda_1\|\mathbf{\Psi}^T\mathbf{K}(\mathbf{I} - \mathbf{C})\|_F^2 \\ + \lambda_2\|\mathbf{Y}(\mathbf{I} - \mathbf{\Psi}\mathbf{\Psi}^T\mathbf{K})\|_F^2,\end{aligned} \tag{11}$$

where $\mathbf{K} = \mathbf{Y}^T\mathbf{Y}$. The equality constraint now becomes

$$\mathbf{P}\mathbf{P}^T = \mathbf{\Psi}^T\mathbf{K}\mathbf{\Psi} = \mathbf{I}. \tag{12}$$

As a result, the optimization problem (8) can be re-written as

$$\begin{aligned}[\mathbf{\Psi}^*, \mathbf{C}^*] = \arg\min_{\mathbf{\Psi},\mathbf{C}} \mathcal{J}_1(\mathbf{C}) + \mathcal{J}_2(\mathbf{\Psi}, \mathbf{C}, \mathbf{Y}) \\ \text{subject to} \quad \mathbf{\Psi}^T\mathbf{K}\mathbf{\Psi} = \mathbf{I}, \text{diag}(\mathbf{C}) = \mathbf{0}.\end{aligned} \tag{13}$$

This formulation allows the update of $\mathbf{P}$ via $\mathbf{\Psi}$. Furthermore, as will become apparent later, this form of the cost function makes it easier to extend the algorithm to non-linear manifolds using kernel methods. We can solve the above optimization problem by optimizing over $\mathbf{\Psi}$ and $\mathbf{C}$ iteratively.

B. Update step for $\mathbf{\Psi}$

In this step, we assume that $\mathbf{C}$ is fixed. So $\mathcal{J}_1$ can be removed and the following problem needs to be solved

$$\lambda_1\|\mathbf{\Psi}^T\mathbf{K}(\mathbf{I}-\mathbf{C})\|_F^2+\lambda_2\|\mathbf{Y}(\mathbf{I}-\mathbf{\Psi}\mathbf{\Psi}^T\mathbf{K})\|_F^2 \quad \text{subject to } \mathbf{\Psi}^T\mathbf{K}\mathbf{\Psi}=\mathbf{I}. \tag{14}$$

The cost function can be expanded as follows

$$\begin{aligned}&\text{trace}\left(\lambda_1(\mathbf{I}-\mathbf{C})(\mathbf{I}-\mathbf{C})^T\mathbf{K}^T\mathbf{Q}^T\mathbf{K}\right)\\&+\text{trace}\left(\lambda_2(\mathbf{K}-2\mathbf{K}^T\mathbf{Q}^T\mathbf{K}+\mathbf{K}^T\mathbf{Q}^T\mathbf{K}\mathbf{Q}\mathbf{K})\right),\end{aligned} \tag{15}$$

where $\mathbf{Q}=\mathbf{\Psi}\mathbf{\Psi}^T\in\mathbb{R}^{N\times N}$. The constraint $\mathbf{\Psi}^T\mathbf{K}\mathbf{\Psi}=\mathbf{I}$ leads to the new constraint

$$\mathbf{\Psi}\mathbf{\Psi}^T\mathbf{K}\mathbf{\Psi}\mathbf{\Psi}^T=\mathbf{\Psi}\mathbf{\Psi}^T$$

or $\mathbf{Q}\mathbf{K}\mathbf{Q}^T=\mathbf{Q}$. The objective function (15) can be further simplified as

$$\text{trace}\left(\left(\lambda_1(\mathbf{I}-\mathbf{C})(\mathbf{I}-\mathbf{C})^T-\lambda_2\mathbf{I}\right)\mathbf{K}^T\mathbf{Q}^T\mathbf{K}\right), \tag{16}$$

where we have made use of the equality constraint and used the fact that trace$(\mathbf{K})$ is constant. Using the eigen decomposition of $\mathbf{K}=\mathbf{V}\mathbf{S}\mathbf{V}^T$, we get

$$\mathbf{K}^T\mathbf{Q}^T\mathbf{K}=\mathbf{V}\mathbf{S}^{\frac{1}{2}}\mathbf{M}\mathbf{M}^T\mathbf{S}^{\frac{1}{2}}\mathbf{V}^T,$$

where $\mathbf{M}=\mathbf{S}^{\frac{1}{2}}\mathbf{V}^T\mathbf{\Psi}$. As a result, (16) can be rewritten as

$$\text{trace}\left(\mathbf{M}^T\mathbf{S}^{\frac{1}{2}}\mathbf{V}^T\left(\lambda_1(\mathbf{I}-\mathbf{C})(\mathbf{I}-\mathbf{C})^T-\lambda_2\mathbf{I}\right)\mathbf{V}\mathbf{S}^{\frac{1}{2}}\mathbf{M}\right).$$

Using the fact that, $\mathbf{\Psi}^T\mathbf{K}\mathbf{\Psi}=\mathbf{M}^T\mathbf{M}$ and with the following change of variable

$$\mathbf{\Delta}=\mathbf{S}^{\frac{1}{2}}\mathbf{V}^T\left(\lambda_1(\mathbf{I}-\mathbf{C})(\mathbf{I}-\mathbf{C})^T-\lambda_2\mathbf{I}\right)\mathbf{V}\mathbf{S}^{\frac{1}{2}},$$

we arrive at the following optimization problem, which is equivalent to (14)

$$\mathbf{M}^*=\min_{\mathbf{M}}\ \text{trace}\left(\mathbf{M}^T\mathbf{\Delta}\mathbf{M}\right)\ \text{ such that }\ \mathbf{M}^T\mathbf{M}=\mathbf{I}. \tag{17}$$

Problem (17) is the classical minimum eigenvalue problem whose solution is given by the ℓ eigenvectors associated with the first ℓ smallest eigenvalues of $\mathbf{\Delta}$ [34]. Once the optimal $\mathbf{M}^*$ is found, the optimal $\mathbf{\Psi}^*$ can be recovered as

$$\mathbf{\Psi}^*=\mathbf{V}\mathbf{S}^{-\frac{1}{2}}\mathbf{M}^*.$$

Hence, we have proved the following proposition:

Proposition 2: The optimal solution of (13) when $\mathbf{C}$ is fixed is

$$\mathbf{\Psi}^*=\mathbf{V}\mathbf{S}^{-\frac{1}{2}}\mathbf{M}^*, \tag{18}$$

where $\mathbf{V}$ and $\mathbf{S}$ come from the eigen decomposition of $\mathbf{K}=\mathbf{V}\mathbf{S}\mathbf{V}^T$, and $\mathbf{M}^*\in\mathbb{R}^{N\times t}$ is the optimal solution of the following minimum eigenvalues problem

$$\mathbf{M}^*=\min_{\mathbf{M}}\ \text{trace}\left(\mathbf{M}^T\mathbf{\Delta}\mathbf{M}\right)\ \text{ such that }\ \mathbf{M}^T\mathbf{M}=\mathbf{I}. \tag{19}$$

where

$$\mathbf{\Delta}=\mathbf{S}^{\frac{1}{2}}\mathbf{V}^T\left(\lambda_1(\mathbf{I}-\mathbf{C})(\mathbf{I}-\mathbf{C})^T-\lambda_2\mathbf{I}\right)\mathbf{V}\mathbf{S}^{\frac{1}{2}}.$$

C. Update step for $\mathbf{C}$

For a fixed $\mathbf{\Psi}$, we have to solve the following problem to obtain $\mathbf{C}$:

$$\min_{\mathbf{C}}\mathcal{J}_1(\mathbf{C})+\lambda_1\|\mathbf{B}-\mathbf{B}\mathbf{C}\|_F^2\ \text{ such that }\ \text{diag}(\mathbf{C})=\mathbf{0}, \tag{20}$$

where $\mathbf{B}=\mathbf{\Psi}^T\mathbf{K}$. Depending on the choice of $\mathcal{J}_1$, this problem can be solved in many different ways.

1) Sparse Representation: In the case when sparsity is enforced, we have to solve the following problem

$$\min_{\mathbf{C}}\|\mathbf{C}\|_1+\lambda_1\|\mathbf{B}-\mathbf{B}\mathbf{C}\|_F^2\ \text{ such that }\ \text{diag}(\mathbf{C})=\mathbf{0}. \tag{21}$$

This problem is the same as the SSC problem, except that the data matrix $\mathbf{Y}$ is replaced by the $\mathbf{B}$ matrix. Therefore, it can be solved by the ADMM method [30], [17]. We call the resulting algorithm Latent Space Sparse Subspace Clustering (LS3C).

2) Low-rank Representation: When low-rank representation is sought, $\mathcal{J}_1(\mathbf{C})=\|\mathbf{C}\|_*$ and the following optimization problem needs to be solved

$$\min_{\mathbf{C}}\|\mathbf{C}\|_*+\lambda_1\|\mathbf{B}-\mathbf{B}\mathbf{C}\|_F^2. \tag{22}$$

From Theorem 2, the closed-form solution to this problem can be computed from the SVD of $\mathbf{B}$. We call the resulting algorithm Latent Space Low-rank Representation (LSLRR) based clustering.

3) Sparse and Low-rank Representation: One can also look for a representation that is simultaneously sparse and low-rank by solving the following problem

$$\min_{\mathbf{C}}\|\mathbf{C}\|_*+\lambda\|\mathbf{C}\|_1+\lambda_1\|\mathbf{B}-\mathbf{B}\mathbf{C}\|_F^2 \quad \text{such that } \ \text{diag}(\mathbf{C})=\mathbf{0}. \tag{23}$$

This problem is similar to the LRSSC problem which can be efficiently solved using the ADMM method [30], [23]. We refer to this method as the Latent Space Low-rank and Sparse Subspace Clustering (LSLRSSC).

IV. Non-Linear Latent Space Subspace Clustering

In many subspace clustering problems, projecting the original features onto a latent space may not be good enough due to non-linearity in data. One approach to dealing with nonlinear manifolds is to transform the data into a high-dimensional feature space using kernel methods. In particular, kernel-based representations have been exploited before in the context of compressed sensing [35], sparse coding [36], dictionary learning [37], [33] and low-rank representation [38]. It has been shown that the non-linear mapping using the kernel trick can group the data with the same distribution and make them linearly separable. The resulting sparse and low-rank representation can provide better clustering.

Let $\Phi:\mathbb{R}^D\to\mathcal{H}$ be a mapping from the input space to the reproducing kernel Hilbert space $\mathcal{H}$. The non-linear mapping $\mathcal{P}$ can be characterized by a compact linear operator $\boldsymbol{\mathcal{P}}:\mathcal{H}\to$

$\mathbb{R}^t$. Let $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{N\times N}$ be a positive semidefinite kernel Gram matrix whose elements are computed as

$$\begin{aligned}[\boldsymbol{\mathcal{K}}(\mathbf{Y},\mathbf{Y})]_{i,j} &= [\langle \Phi(\mathbf{Y}), \Phi(\mathbf{Y})\rangle_{\mathcal{H}}]_{i,j} \\ &= \Phi(\mathbf{y}_i)^T\Phi(\mathbf{y}_j) \\ &= \kappa(\mathbf{y}_i,\mathbf{y}_j),\end{aligned} \tag{24}$$

where $\kappa : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is the kernel function and

$$\Phi(\mathbf{Y}) = [\Phi(\mathbf{y}_1), \Phi(\mathbf{y}_2), \cdots, \Phi(\mathbf{y}_N)].$$

Some commonly used kernels include polynomial kernels

$$\kappa(\mathbf{x},\mathbf{y}) = \left(\langle \mathbf{x},\mathbf{y}\rangle + a\right)^b$$

and Gaussian kernels

$$\kappa(\mathbf{x},\mathbf{y}) = \exp\left(-\sigma\|\mathbf{x}-\mathbf{y}\|^2\right),$$

where a, b and σ are the parameters of the kernel functions.

With the above definitions, the latent space subspace clustering methods can be made non-linear by writing the cost functions as follows

$$\begin{aligned}\mathcal{J}_1(\mathbf{C}) + \lambda_1\|\boldsymbol{\mathcal{P}}\Phi(\mathbf{Y}) - \boldsymbol{\mathcal{P}}\Phi(\mathbf{Y})\mathbf{C}\|_F^2 \\ + \lambda_2\|\Phi(\mathbf{Y}) - \boldsymbol{\mathcal{P}}^T\boldsymbol{\mathcal{P}}\Phi(\mathbf{Y})\|_F^2.\end{aligned} \tag{25}$$

This formulation is the same as that in (8) except that $\mathbf{Y}$ is now replaced by $\Phi(\mathbf{Y})$. Furthermore, similar to Proposition 1, it can be shown that the optimal projection takes the following form

$$\boldsymbol{\mathcal{P}}^* = \boldsymbol{\Psi}^T\Phi(\mathbf{Y})^T. \tag{26}$$

As a result, we get the following cost function

$$\begin{aligned}\mathcal{J}_1(\mathbf{C}) + \lambda_1\|\boldsymbol{\Psi}^T\boldsymbol{\mathcal{K}}(\mathbf{I}-\mathbf{C})\|_F^2 \\ + \lambda_2\text{trace}\left((\mathbf{I}-\boldsymbol{\Psi}\boldsymbol{\Psi}^T\boldsymbol{\mathcal{K}})^T\boldsymbol{\mathcal{K}}(\mathbf{I}-\boldsymbol{\Psi}\boldsymbol{\Psi}^T\boldsymbol{\mathcal{K}})\right)\end{aligned} \tag{27}$$

and the constraint $\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{P}}^T = \mathbf{I}$ becomes $\boldsymbol{\Psi}^T\boldsymbol{\mathcal{K}}\boldsymbol{\Psi} = \mathbf{I}$. This optimization problem can be solved in the same way as the linear case. The update steps for $\boldsymbol{\Psi}$ and $\mathbf{C}$ remain the same except that $\mathbf{K}$ is now replaced with $\boldsymbol{\mathcal{K}}$. We refer to the nonlinear versions of the LS3C, LSLRR and LSLRSSC algorithms as NLS3C, NLSLRR and NLSLRSSC, respectively.

Note that the dimension of the output space is upper bounded by the number of training samples. Both the linear and non-linear methods for finding the sparse and low-rank coefficient matrices in the latent space along with the projection matrix are summarized in Algorithm 1.

Similar to the SSC, LRR and LRSSC methods, once the coefficient matrix $\mathbf{C}$ is found, spectral clustering is applied on the affinity matrix $\mathbf{W} = |\mathbf{C}|+|\mathbf{C}|^T$ to obtain the segmentation of the data in the low-dimensional latent space. The proposed latent space subspace clustering methods are summarized in Algorithm 2.

Note that a learned transformation-based approach for subspace clustering and classification was recently proposed in [20], but we differ from this work in a few key areas. Unlike [20], our method does not require the leaned projection to be $D \times D$. Furthermore, the optimization approach of [20] requires the cluster labels of each point. As a result they rely on standard subspace clustering methods to assign points to clusters before learning the transformation. In our formulation, we jointly find the optimal transformation and the sparse and/or low-rank representation. Furthermore, our method is applicable to LRR, SSC and LRSSC algorithms and can deal with data that lie in a union of affine subspaces. Also, we present nonlinear extensions of the proposed algorithms using the kernel trick.

Algorithm 1: Simultaneous dimension reduction and subspace clustering for both linear and non-linear cases.

Input: Kernel matrix $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{N\times N}, \lambda_1, \lambda_2, \lambda$.
Initialization:
- Set iteration $J = 1$. Perform eigen decomposition $\boldsymbol{\mathcal{K}} = \mathbf{V}\mathbf{S}\mathbf{V}^T$.
- Set $\boldsymbol{\Psi} = \mathbf{V}(:,\mathcal{I})$, where $\mathcal{I}$ is the index set of the d largest eigenvalues of $\boldsymbol{\mathcal{K}}$.
Stage 1: Fix $\boldsymbol{\Psi}$ and update $\mathbf{C}$
- Compute $\mathbf{B} = \boldsymbol{\Psi}^T\boldsymbol{\mathcal{K}}$.
- NLS3C: Solve the optimization problem (21) to obtain $\mathbf{C}$.
- NLSLRR: Solve the optimization problem (22) to obtain $\mathbf{C}$.
- NLSLRSSC: Solve the optimization problem (23) to obtain $\mathbf{C}$.
Stage 2: Fix $\mathbf{C}$ and update $\boldsymbol{\Psi}$
- Compute $\boldsymbol{\Delta} = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^T\left(\lambda_1(\mathbf{I}-\mathbf{C})(\mathbf{I}-\mathbf{C})^T - \lambda_2\mathbf{I}\right)\mathbf{V}\mathbf{S}^{\frac{1}{2}}$.
- Perform eigen decomposition of $\boldsymbol{\Delta} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$.
- Set $\mathbf{M} = \mathbf{U}(:,\mathcal{I}_J)$, where $\mathcal{I}_J$ is the index set of the d smallest eigenvalues of $\boldsymbol{\Delta}$.
- Update $\boldsymbol{\Psi} = \mathbf{V}\mathbf{S}^{-\frac{1}{2}}\mathbf{M}$.
- Increment $J = J+1$. Repeat from stage 1 until stopping conditions reached.
Output: $\mathbf{C}$ and $\boldsymbol{\Psi}$.

Algorithm 2: Latent Space Subspace Clustering for both linear and non-linear cases.

Input: Kernel matrix $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{N\times N}, \lambda_1, \lambda_2, \lambda$.
Algorithm:
- Apply Algorithm 1 to find the sparse coefficient matrix $\mathbf{C}$.
- Normalize the columns of $\mathbf{C}$ as $\mathbf{c}_i \leftarrow \frac{\mathbf{c}_i}{\|\mathbf{c}_i\|_\infty}$.
- Form a similarity graph with N nodes and set the weights on the edges between the nodes by $\mathbf{W} = |\mathbf{C}| + |\mathbf{C}|^T$.
- Apply spectral clustering to the similarity graph.
Output: Cluster labels for all signals.

V. Experimental Results

In this section, we evaluate our proposed methods on both synthetic and real datasets. In particular, the effectiveness of our linear and non-linear subspace clustering methods is evaluated on two computer vision tasks: motion segmentation and hand-written digit clustering. We compare our methods with several state-of-the-art subspace clustering algorithms such as SSC [17], LRR [15], Low-Rank Subspace Clustering (LRSC) [16], Local Subspace Affinity (LSA) [10] and Spectral Curvature Clustering (SCC) [8]. For all the experiments, we set the maximum number of iteration in our Algorithm 1 to $J = 3$. We set $\lambda_1 = \lambda_2 = 50$. All the experiments are done on an OS X system with 2.6 GHz Intel Core i7 processor using Matlab. Subspace clustering error is used to measure the performance of different algorithms. It is defined as

$$\text{subspace clustering error} = \frac{\#\text{of misclassified points}}{\text{total}\#\text{of points}} \times 100.$$

A. Synthetic Data

In this section, we generate a synthetic data to study the performance of LS3C and NLS3C when the data in each

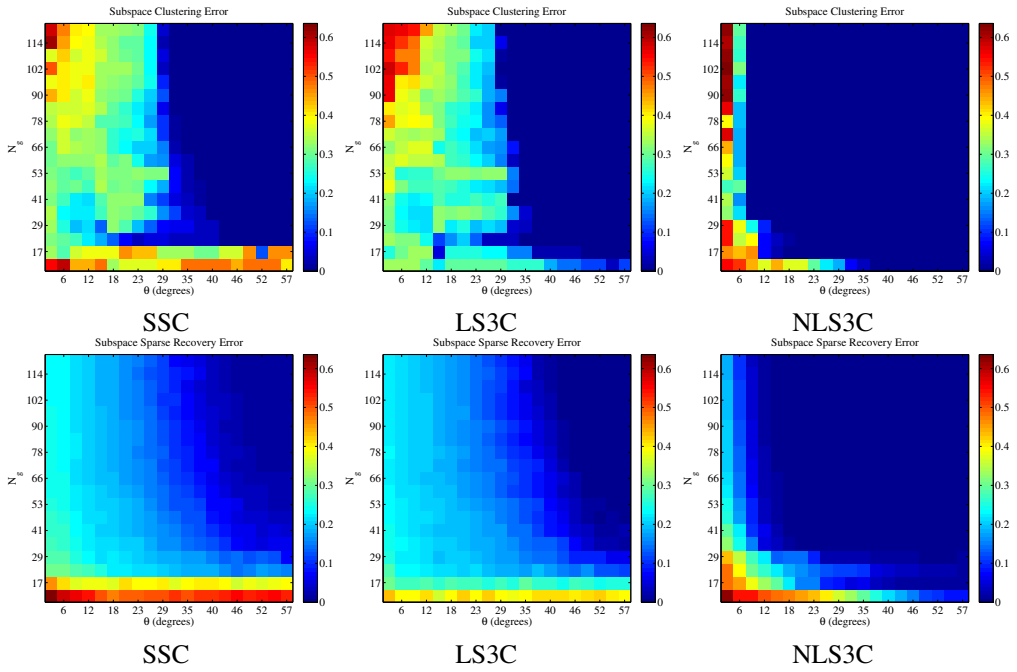


Fig. 2: Subspace clustering errors and subspace-sparse recovery errors for three randomly generated disjoint subspaces with different smallest principle angles and different number of points.

subspace and the smallest principle angle between subspaces are small. We follow the same experimental setting as in [17]. We consider $n = 3$ subspaces of dimension $d = 3$ embedded in $D = 50$ dimensional space. We generate the bases $\{\mathbf{T}_i \in \mathbb{R}^{D\times d}\}_{i=1}^{3}$ such that rank $([\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3]) = 2d$. Also, the subspaces are generated such that $\theta_{12} = \theta_{23} = \theta$. Furthermore, we generate the same number of points, N_g, in each subspace at random and change the value of N_g.

For a fixed value of d, we change the minimum angle between subspaces, θ, as well as the number of points in each subspace N_g. For each pair of (θ, N_g), we compute the subspace clustering error. Since the performance of LS3C and NLS3C methods are based on how well the sparse coefficients are found, we also calculate the subspace sparse recovery error. For the data points $\{\mathbf{y}_i\}_{i=1}^{3N_g}$, the sparse recovery error E_{SR} is given by

$$E_{SR} = \frac{1}{3N_g}\sum_{i=1}^{3N_g}\left(1 - \frac{\|\mathbf{c}_{iq_i}\|_1}{\|\mathbf{c}_i\|_1}\right),$$

where $\mathbf{c}_i^T = [\mathbf{c}_{i1}{}^T, \mathbf{c}_{i2}{}^T, \mathbf{c}_{i3}{}^T]$ represents the sparse coefficients of $\mathbf{y}_i \in \mathcal{S}_{q_i}$ and $\mathbf{c}_{ij}$ corresponds to the points in $\mathcal{S}_j$.

We vary the smallest principle angle between subspaces and the number of points in each subspace as $\theta \in [6, 60]$ and $N_g \in [d+1, 20d]$, respectively. For each pair (θ, N_g), we calculate the average subspace clustering error as well as the average E_{SR} over 20 trials. In each trial we randomly generate data points and subspaces. Results of this experiment are shown in Figure 2. When θ and N_g decrease both the sparse recovery and clustering errors of all the methods increase. Also, the clustering error is highly dependent on the sparse recovery error and both errors follow the same pattern. In other words, clustering results are highly dependent on how well the sparse coefficients are recovered. By comparing the decay of errors, one can see that in the case where both θ and N_g are small, our methods perform better than the SSC method. The error decays faster in the case of LS3C and NLS3C than SSC. This can be explained by the fact that our method finds the projection directly from data and preserves the sparse structure of data in the latent space. In this experiment, for NLS3C we used a polynomial kernel with parameters $b = 3$ and $a = 0$.

B. Motion Segmentation

In motion segmentation, the idea is to segment a video sequence into multiple spatiotemporal regions corresponding to different rigid body motions. Suppose that we have tracked N feature points over F frames in a video sequence, $\{\mathbf{x}_{ij}\}$, where $i = 1, \cdots, N$ and $j = 1 \cdots, F$. Each feature trajectory $\mathbf{y}_i \in \mathbb{R}^{2F}$ is obtained by stacking the feature points in the video, i.e

$$\mathbf{y}_i^T = [\mathbf{x}_{1i}^T, \mathbf{x}_{2i}^T, \cdots, \mathbf{x}_{Fi}^T].$$

Then, the objective is to separate these feature trajectories according to their motions. It has been shown that trajectories of a general rigid motion under affine projection span a $4n$-dimensional linear subspace [2]. In other words, feature trajectories of n rigid motions lie in a union of n-dimensional subspaces of $\mathbb{R}^{2F}$. Hence, the problem of clustering the trajectories according to the different motion is equivalent to the problem of clustering affine subspaces.

We apply our non-linear subspace clustering frameworks to the Hopkins155 motion segmentation database [26]. The dataset contains 155 video sequences where 120 video sequences contain 2 motions and 35 video sequences have 3 motions. For each sequence, a tracker is used to extract the point trajectories and the outliers are extracted manually [26]. On average, each sequence of 2 motions has 266 feature

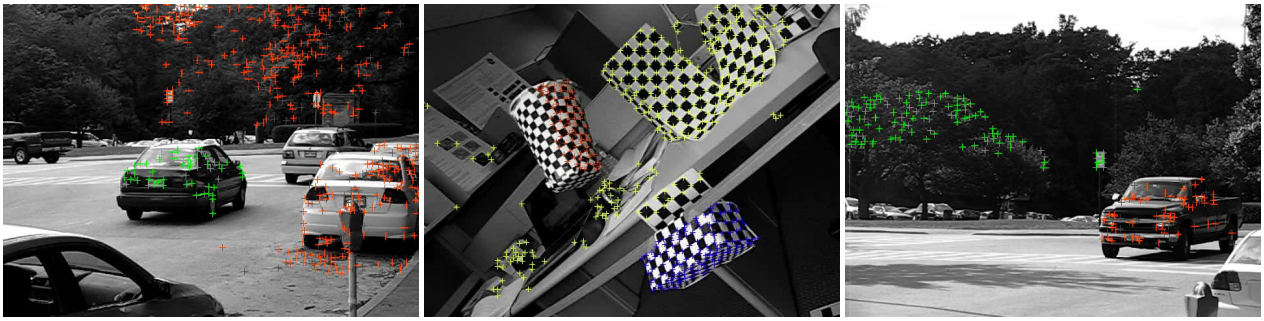


Fig. 3: Sample video frames from the Hopkins155 dataset.

trajectories and 30 frames and each sequence of 3 motions has 398 feature trajectories and 29 frames. Sample frames from this dataset are shown in Fig. 3. A polynomial kernel with parameters $a = 0.3$ and $b = 2$ is used in these experiments.

Table I compares the performance of different methods. For the subspace clustering algorithms other than the proposed methods, the data is first projected onto the $4n$-dimensional subspace using PCA [17]. As can be seen from this table, on average the proposed latent space subspace clustering methods perform better than SSC, LRR and LRSSC. They are able to learn the projection directly from the data better than PCA for clustering. The LS3C method performs the best on both 2 motion and 3 motion sequences. The proposed non-linear subspace clustering methods also obtain small clustering errors compared to the other competitive subspace clustering algorithms.

In the second set of experiments with the Hopkins155 dataset, we study the performance of different methods as we vary the subspace dimensions. We project the data onto the following dimensional subspaces: $\{2n, 6n, 8n, 10n\}$. For the LRR, SSC and LRSSC methods, we project the data onto the low-dimensional space using random projections and PCA. Random projections have been used for dimensionality reduction in many sparsity-based algorithms [39], [11] and they have been shown to preserve the sparsity of data provided certain conditions are met [40]. Let $\mathbf{P}$ be an $t \times D$ random matrix with $t \leq D$ such that each entry $p_{i,j}$ of $\mathbf{P}$ is an independent realization of q, where q is a random variable on a probability measure space (Ω, ρ). It has been shown that given any set of points Λ, the following are some of the matrices that provide the sparsest solution via ℓ_1 minimization problem provided that enough measurements are taken [40]:

- RP1: $t \times D$ random matrix $\mathbf{P}$ whose entries $p_{i,j}$ are independent realizations of Gaussian random variables $p_{i,j} \sim \mathcal{N}\left(0, \frac{1}{t}\right)$.
- RP2: Independent realizations of ± 1 Bernoulli random variables

$$p_{i,j} \doteq \begin{cases} +1/\sqrt{t}, & \text{with probability } \frac{1}{2} \\ -1/\sqrt{t}, & \text{with probability } \frac{1}{2}. \end{cases}$$

We use both RP1 and RP2 to project the data points onto a low-dimensional space. The average clustering error results are summarized in Table II. It can be seen from this table that NLS3C, NLSLRR and NLSLRSSC methods consistently outperform their linear counterparts in all dimensions. It is also interesting to note that the performance of SSC, LRR and LRSSC varies depending on the projection matrix used for dimensionality reduction. In other words, features are important for SSC, LRR and LRSSC. In contrast, our method automatically learns the features directly from the data and consistently performs better than LRR, SSC and LRSSC.

C. Rotated Hand-written Digit Clustering

The rotated MNIST benchmark [41] contains gray scale images of hand-written digits of size 28×28 pixels. The images were originally taken from the MNIST dataset introduced in [42], and transformed in several ways to create more challenging classification problems. In the first dataset, called the *mnist-rot*, digits are rotated by random angles generated uniformly between 0 and 2π radians. The second dataset, called the *mnist-rot-back-image*, is created by inserting random backgrounds into *mnist-rot* dataset. The *mnist-back-rand* dataset is created by inserting random backgrounds in the original MNIST digit images. For all 3 datasets, there are 10000, 2000, and 50000 images for training, validation, and testing, respectively. Figure 4 shows sample images from the above datasets.

Fig. 4: Sample digits from the rotated MNIST dataset. (a) Digits with random rotations, (b) Digits with random rotations and image backgrounds, (c) Digits with random backgrounds.

We evaluate the clustering performance of various methods on this challenging dataset. It was shown in [3] that handwritten digits with some variations lie on 12-dimensional subspaces. Hence, n hand-written digits can be modeled as data points lying close to a union of 12-dimensional subspaces. Since this dataset contains a large amount of samples (about 62,000 samples), we only use samples from the training and the validation sets (12,000 samples) for clustering. In

Algorithms (2 Motions)	SSC	LRR	LRSSC	SCC	LSA	LRSC	LS3C	NLS3C	LSLRR	NLSLRR	LSLRSSC	NLSLRSSC
Mean	1.83	3.41	3.07	3.04	3.61	2.57	**1.62**	1.79	2.55	2.56	3.34	3.28
Median	**0.00**	**0.00**	**0.00**	**0.00**	0.51	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Algorithms (3 Motions)	SSC	LRR	LRSSC	SCC	LSA	LRSC	LS3C	NLS3C	LSLRR	NLSLRR	LSLRSSC	NLSLRSSC
Mean	4.40	4.86	6.68	7.91	7.65	6.62	**4.38**	4.89	7.04	5.29	8.89	8.34
Median	**0.56**	1.47	0.81	1.14	1.27	1.76	**0.56**	0.85	2.20	1.22	4.99	4.01
Algorithms (All)	SSC	LRR	LRSSC	SCC	LSA	LRSC	LS3C	NLS3C	LSLRR	NLSLRR	LSLRSSC	NLSLRSSC
Mean	2.41	3.74	5.68	4.14	4.52	3.47	**2.31**	2.56	3.67	3.29	4.89	4.51
Median	**0.00**	**0.00**	**0.00**	**0.00**	0.57	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**

TABLE I: Clustering errors on the Hopkins155 dataset with the $4n$-dimensional data points. The top performing method in each experiment is shown in boldface.

Algorithms (2 Motions)	SSC (PCA)	LRR (PCA)	LRSSC (PCA)	SSC (RP2)	SSC (RP1)	LRR (RP2)	LRR (RP1)	LRSSC (RP2)	LRSSC (RP1)	LS3C	NLS3C	LSLRR	NLSLRR	LSLRSSC	NLSLRSSC
$2n$-Dim	3.33	7.09	13.75	5.08	4.29	13.91	12.73	13.76	13.19	**3.23**	3.86	7.09	4.76	8.70	4.57
$6n$-Dim	2.34	4.21	12.25	2.40	2.65	8.35	8.44	12.01	12.11	**2.31**	2.57	3.70	3.39	5.67	3.99
$8n$-Dim	2.33	4.20	10.09	2.60	2.92	7.79	7.36	11.70	11.01	**2.29**	2.57	3.69	3.38	5.67	3.98
$10n$-Dim	2.33	4.19	9.08	2.40	2.59	7.90	7.74	10.41	10.12	**2.29**	2.57	3.69	3.38	5.67	3.98

TABLE II: Average clustering errors on the Hopkins155 dataset with different dimensional data points. The top performing method in each experiment is shown in boldface.

particular, we select 10 samples per digit and generate a small subset containing 100 samples from 10 digits. We use these samples for clustering and repeat the process 120 times so that all the samples from the training and the validation sets are used for clustering.

We report the average clustering performances of different methods in Table III. As can be seen from this table, in all cases, NLS3C performances compare favorably to the state-of-the-art. By non-linearly projecting the data, we are able to capture the compact structure of data that is more robust against noise. Polynomial kernel with $a = 1, b = 4$ is used in this experiment. The performance of LS3C is also comparable to that of SSC. Even though LRR, SSC and LRSSC methods can separate the background and remove noise from the data, they do not perform well on this dataset. This is the case because these methods can not find the sparse and low-rank representation of the samples when the data contains random rotations. In contrast, our non-linear projection learns the rotation mapping directly from the data. Figure 5 displays the transformations learned by our methods on the *mnist-rot* dataset. Each subplot of Figure 5 corresponds to a row of the matrix $\mathbf{P} = \mathbf{\Psi}^T\mathbf{Y}^T$. They have a strong similarity to circular harmonic functions, thus, can capture more rotational invariant features. These transformations make a good sense given that the dataset consists of a lot of variations along the circular direction.

Two most computationally heavy steps of our methods are the computation of sparse and/or low-rank coefficients and spectral clustering. The average times are shown in the last row of Table III. On average NLS3C and NLSLRSSC methods take about 13 seconds to cluster 100 digits of size 28×28, whereas SSC and LRSSC methods take about 14 seconds. The LSLRR and NLSLRR methods are the most computationally heavy methods compared to LS3C and LSLRSSC because they require taking the SVD of large matrices in each iteration of the algorithms. Figure 6(a)-(c) show the cost functions with iterations for the proposed methods. It can be seen that both the linear and non-linear algorithms converge in a few iterations.

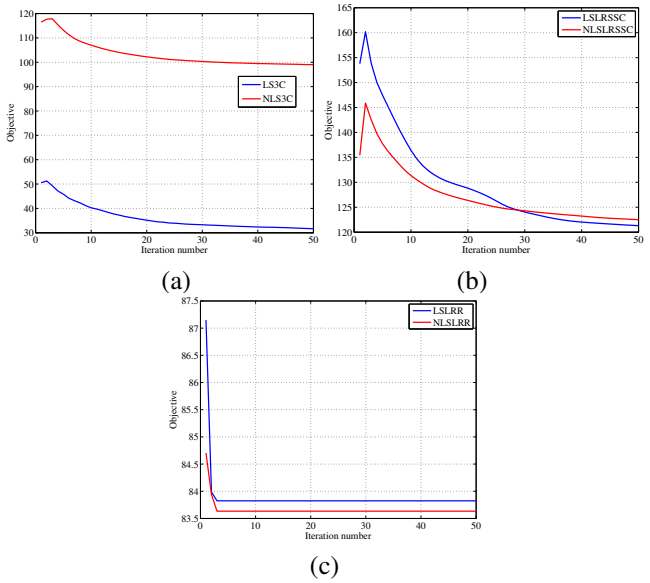


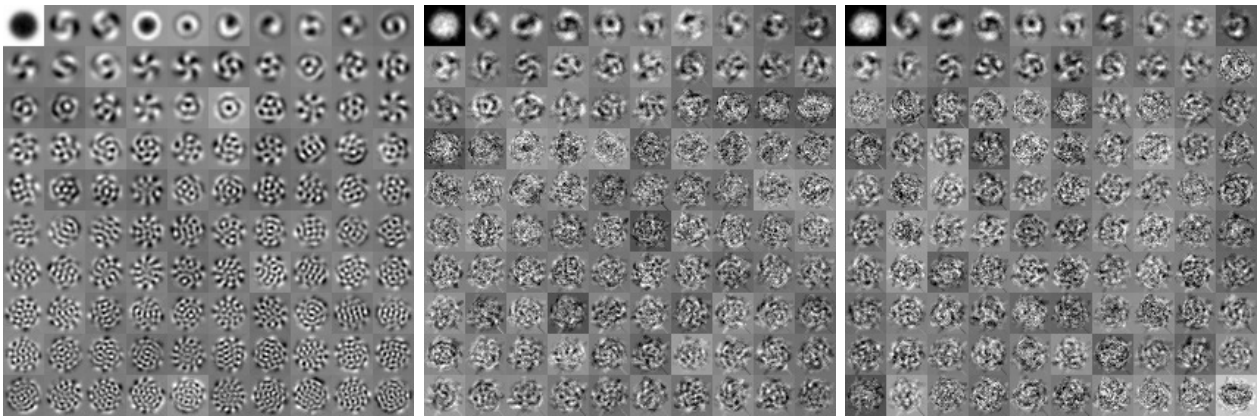
Fig. 6: The objective function value as a function of iteration number for the experiments with the rotated MNIST dataset. (a) LS3C and NLS3C. (b) LSLRSSC and NLSLRSSC. (c) LSLRR and NLSLRR.

VI. Conclusion

We have proposed three simultaneous dimensionality reduction and sparse and low-rank representation methods in the low-dimensional latent space for SSC, LRR and LRSSC.

Dataset	SSC	LRR	LRSSC	LS3C	NLS3C	LSLRR	NLSLRR	LSLRSSC	NLSLRSSC
(a)	67.75	75.48	68.38	67.62	**66.49**	67.56	68.79	66.90	68.33
(b)	74.31	80.06	74.08	74.30	**72.38**	75.83	74.90	73.48	73.48
(c)	58.59	77.75	62.28	58.68	**54.63**	80.36	66.10	62.73	56.23
Avg. Time (sec)	13.86	**12.56**	13.89	13.06	13.10	76.25	76.28	13.41	13.42

TABLE III: Average clustering errors on the rotated MNIST datasets: (a) *mnist-rot*, (b) *mnist-rot-back-image*, (c) *mnist-back-rand*. The top performing method in each experiment is shown in boldface.



(a) (b) (c)

Fig. 5: Example of transformations learned by (a) the LS3C method (b) the LSLRSSC method and (c) the LSLRR method from the rotated MNIST dataset.

Efficient optimization algorithms are presented. Furthermore, the methods are kernalized so that they can deal with non-linear manifolds. Through extensive clustering experiments on several datasets, it was shown that the proposed methods are robust and can perform significantly better than many state-of-the-art subspace clustering methods.

ACKNOWLEDGMENT

The work of VMP was partially supported by an Office of Naval Research grant N00014-12-1-0124. The work of RV was partially supported by National Science Foundation grant 11-1218709.

REFERENCES

[1] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 218–233, 2003.

[2] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *International Journal of Computer Vision*, vol. 29, no. 3, pp. 159–179, 1998.

[3] T. Hastie and P. Y. Simard, "Metrics and models for handwritten character recognition," *Statistical Science*, vol. 13, no. 1, pp. 54–65.

[4] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.

[5] 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.

[6] T. Zhang, A. Szlam, and G. Lerman, "Median k-flats for hybrid linear modeling with many outliers," in *Workshop on Subspace Methods*, 2009.

[7] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, *International Journal of Computer Vision*, vol. 100, no. 3, pp. 217–240, 2012.

[8] G. Chen and G. Lerman, "Spectral curvature clustering (scc)," *International Journal of Computer Vision*, vol. 81, no. 3, pp. 317–330, 2009.

[9] A. Goh and R. Vidal, "Segmenting motions of different types by unsupervised manifold clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[10] J. Yan and M. Pollefeys, "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate," in *European Conf. on Computer Vision*, 2006, p. 94106.

[11] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2790–2797.

[12] M. A. Fischler and R. C. Bolles, "Ransac random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 26, pp. 381–395, 1981.

[13] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (gpca)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1–15, 2005.

[14] K. Kanatani, "Motion segmentation by subspace separation and model selection," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2001, pp. 586–591.

[15] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *International Conference on Machine Learning*, 2010.

[16] 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.

[17] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[18] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[19] R. Vidal and P. Favaro, "Low rank subspace clustering (LRSC)," *Pattern Recognition Letters*, 2013.

[20] Q. Qiu and G. Sapiro, "Learning transformations for clustering and classification," *Journal of Machine Learning Research*, 2014.

[21] G. Liu and S. Yan, "Latent low-rank representation for subspace segmentation and feature extraction," in *IEEE International Conference on Computer Vision*, 2011, pp. 1615–1622.

[22] V. M. Patel, H. V. Nguyen, and R. Vidal, "Latent space sparse subspace clustering," in *International Conference on Computer Vision*, 2013.

[23] Y.-X. Wang, H. Xu, and C. Leng, "Provable subspace clustering: When lrr meets ssc," in *Advances in Neural Information Processing Systems 26*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., 2013, pp. 64–72.
[24] V. M. Patel and R. Vidal, "Kernel sparse subspace clustering," in *IEEE International Conference on Image Processing*, 2014.
[25] M. Soltanolkotabi and E. J. Candes, "A geometric analysis of subspace clustering with outliers," *Annals of Statistics*, vol. 40, no. 4, pp. 2195–2238, 2011.
[26] R. Tron, R. Vidal, and A. Ravichandran, "A benchmark for the comparison of 3-d motion segmentation algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
[27] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 791–804, 2012.
[28] D. Zhang, M. Yang, Z. Feng, and D. Zhang, "On the dimensionality reduction for sparse representation based face recognition," in *International Conference on Pattern Recognition*, 2010, pp. 1237–1240.
[29] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Sparse embedding: A framework for sparsity promoting dimensionality reduction." in *European Conference on Computer Vision*, Oct. 2012, pp. 414–427.
[30] 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*, vol. 3, no. 1, pp. 1–122, 2011.
[31] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Neural Information Processing Systems*, vol. 2, 2002, pp. 849–856.
[32] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1553–1564, 2010.
[33] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Design of non-linear kernel dictionaries for object recognition," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5123–5135, 2013.
[34] A. H. Sameh and J. A. Wisniewski, "A trace minimization algorithm for the generalized eigenvalue problem," *SIAM J. Numer. Anal.*, vol. 19, no. 6, pp. 1243–1259, 1982.
[35] H. Qi and S. Hughes, "Using the kernel trick in compressive sensing: Accurate signal recovery from fewer measurements," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, may 2011, pp. 3940 –3943.
[36] L. Zhang, W.-D. Zhou, P.-C. Chang, J. Liu, Z. Yan, T. Wang, and F.-Z. Li, "Kernel sparse representation-based classifier," *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 1684 –1695, april 2012.
[37] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Kernel dictionary learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 2021–2024.
[38] J. Wang, V. Saligrama, and D. Castanon, "Structural similarity and distance in learning," in *Allerton Conference on Communication, Control, and Computing*, Sept 2011, pp. 744–751.
[39] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 210–227, 2009.
[40] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, Dec. 2008.
[41] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 473–480.
[42] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

APPENDIX

Proof of *Proposition 1*:

Using the orthogonal decomposition of $\mathbf{P}^*$, we have

$$\begin{aligned} &\mathbf{P}^* = \mathbf{P}_{\|} + \mathbf{P}_{\perp}, \\ &\text{where} \quad \mathbf{P}_{\|} = (\mathbf{Y}\mathbf{\Psi})^T \quad \text{and} \quad \mathbf{P}_{\perp}\mathbf{Y} = \mathbf{0} \end{aligned} \tag{27}$$

for some $\mathbf{\Psi} \in \mathbb{R}^{N\times t}$. Using this, we can write the first term of $\mathcal{J}_2(\mathbf{P}, \mathbf{C}.\mathbf{Y})$ as

$$\begin{aligned} \lambda_1\|\mathbf{P}^*\mathbf{Y} - \mathbf{P}^*\mathbf{Y}\mathbf{C}\|_F^2 &= \lambda_1\|\mathbf{P}^*\mathbf{Y}(\mathbf{I}-\mathbf{C})\|_F^2 \\ &= \lambda_1\|(\mathbf{P}_{\|}+\mathbf{P}_{\perp})\mathbf{Y}(\mathbf{I}-\mathbf{C})\|_F^2 \\ &= \lambda_1\|\mathbf{P}_{\|}\mathbf{Y}(\mathbf{I}-\mathbf{C})\|_F^2 \\ &= \text{trace}\left(\lambda_1\mathbf{P}_{\|}\mathbf{Y}(\mathbf{I}-\mathbf{C})(\mathbf{I}-\mathbf{C})^T\mathbf{Y}^T\mathbf{P}_{\|}^T\right). \end{aligned} \tag{28}$$

The second term of $\mathcal{J}_2(\mathbf{P}, \mathbf{C}.\mathbf{Y})$ can be written as

$$\begin{aligned} &\lambda_2\|\mathbf{Y} - \mathbf{P}^T\mathbf{P}\mathbf{Y}\|_F^2 \\ &= \lambda_2\text{trace}\left(\mathbf{Y}^T\mathbf{Y} - \mathbf{Y}^T(\mathbf{P}_{\|}+\mathbf{P}_{\perp})^T(\mathbf{P}_{\|}+\mathbf{P}_{\perp})\mathbf{Y}\right) \\ &= \lambda_2\text{trace}\left(\mathbf{Y}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{P}_{\|}^T\mathbf{P}_{\|}\mathbf{Y}\right) \\ &= \text{trace}\left(\lambda_2\mathbf{Y}^T\mathbf{Y} - \lambda_2\mathbf{P}_{\|}\mathbf{Y}\mathbf{Y}^T\mathbf{P}_{\|}^T\right), \end{aligned} \tag{29}$$

where in the first step of the derivation, we have used the fact that $\mathbf{P}^*\mathbf{P}^{*T} = \mathbf{I}$. Putting equations (27), (28) and (29) and letting $\mathbf{K} = \mathbf{Y}^T\mathbf{Y}$, we get the following objective function

$$\begin{aligned} &\text{trace}(\lambda_2\mathbf{K}) \\ &\quad - \text{trace}\left(\mathbf{P}_{\|}\mathbf{Y}\left(\lambda_2\mathbf{I} - \lambda_1(\mathbf{I}-\mathbf{C})(\mathbf{I}-\mathbf{C})^T\right)\mathbf{Y}^T\mathbf{P}_{\|}^T\right) \\ &\quad = \text{trace}(\lambda_2\mathbf{K}) - \text{trace}\left(\mathbf{\Psi}^T\mathbf{V}\mathbf{S}^{\frac{1}{2}}\tilde{\mathbf{\Delta}}\mathbf{S}^{\frac{1}{2}}\mathbf{V}^T\mathbf{\Psi}\right), \end{aligned} \tag{30}$$

where

$$\tilde{\mathbf{\Delta}} = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^T\left(\lambda_2\mathbf{I} - \lambda_1(\mathbf{I}-\mathbf{C})(\mathbf{I}-\mathbf{C})^T\right)\mathbf{V}\mathbf{S}^{\frac{1}{2}},$$

$\mathbf{K} = \mathbf{V}\mathbf{S}\mathbf{V}^T$. Let $\mathbf{M} = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^T\mathbf{\Psi}$, then (30) can be written as

$$\begin{aligned} &\text{trace}(\lambda_2\mathbf{K}) - \text{trace}\left(\mathbf{M}^T\tilde{\mathbf{\Delta}}\mathbf{M}\right) \\ &\qquad \geq \text{trace}(\lambda_2\mathbf{K}) - \sum_{j=1}^{t}\beta_j, \end{aligned} \tag{31}$$

where β_j is the j-th largest eigenvalue of $\tilde{\mathbf{\Delta}}$. In order for the objective function to achieve its minimum, columns of $\mathbf{M}$ have to be the same with the eigenvectors corresponding to the largest eigenvalues of $\tilde{\mathbf{\Delta}}$. Hence,

$$\mathbf{M}^T\mathbf{M} = \mathbf{\Psi}^T\mathbf{K}\mathbf{\Psi} = \mathbf{P}_{\|}\mathbf{P}_{\|}^T = \mathbf{I} - \mathbf{P}_{\perp}\mathbf{P}_{\perp}^T = \mathbf{I}.$$

In other words, $\mathbf{P}_{\perp} = \mathbf{0}$. Hence, the optimal solution has the following form

$$\mathbf{P}^* = \mathbf{P}_{\|} = \mathbf{\Psi}^T\mathbf{Y}^T.$$

PLACE PHOTO HERE

Vishal M. Patel (M'08) received the B.S. degrees in electrical engineering and applied mathematics (Hons.) and the M.S. degree in applied mathematics from North Carolina State University, Raleigh, NC, USA, in 2004 and 2005, respectively, and the Ph.D. degree in electrical engineering from the University of Maryland College Park, MD, USA, in 2010. He is currently a member of the research faculty with the University of Marylands Institute for Advanced Computer Studies, College Park, MD, USA. His current research interests include signal processing, computer vision, and pattern recognition with applications in biometrics and imaging. He was a recipient of the ORAU Post-Doctoral Fellowship in 2010. He is a member of Eta Kappa Nu, Pi Mu Epsilon, and Phi Beta Kappa.

PLACE PHOTO HERE

Hien Van Nguyen (M'08) received the B.S. and Ph.D. degrees in electrical and computer engineering from the National University of Singapore, Singapore, and University of Maryland at College Park, College Park, MD, USA, in 2007 and 2013, respectively. He is currently a Research Scientist with Siemens Corporate Research, Princeton, NJ, USA. He was awarded a Full Undergraduate Scholarship by the Singapore Ministry of Foreign Affairs. His research interests include computer vision, machine learning, and statistical pattern recognition

PLACE PHOTO HERE

René Vidal (S'01-M'03-SM'11-F'14) received his B.S. degree in Electrical Engineering (valedictorian) from the Pontificia Universidad Católica de Chile in 1997 and his M.S. and Ph.D. degrees in Electrical Engineering and Computer Sciences from the University of California at Berkeley in 2000 and 2003, respectively. He has been a faculty member in the Department of Biomedical Engineering of The Johns Hopkins University since 2004. He was co-editor of the book "Dynamical Vision" and has co-authored more than 180 articles in biomedical image analysis, computer vision, machine learning, hybrid systems, and robotics. Dr. Vidal has been Associate Editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence, the SIAM Journal on Imaging Sciences and the Journal of Mathematical Imaging and Vision, Program Chair for ICCV 2015, CVPR 2014, WMVC 2009 and PSIVT 2007, and Area Chair for MICCAI 2013 and 2014, ICCV 2007, 2011 and 2013, and CVPR 2005 and 2013. He has received many awards for his work including the 2012 J.K. Aggarwal Prize, the 2009 ONR Young Investigator Award, the 2009 Sloan Research Fellowship, the 2005 NFS CAREER Award, and best paper awards at ICCV-3DRR 2013, PSIVT 2013, CDC 2012, MICCAI 2012, CDC 2011 and ECCV 2004. He is a fellow of the IEEE and a member of the ACM and SIAM.