Segmenting Motions of Different Types by Unsupervised Manifold Clustering

Alvina Goh René Vidal
Center for Imaging Science, Johns Hopkins University, Baltimore MD 21218, USA

Abstract

We propose a novel algorithm for segmenting multiple motions of different types from point correspondences in multiple affine or perspective views. Since point trajectories associated with different motions live in different manifolds, traditional approaches deal with only one manifold type: linear subspaces for affine views, and homographic, bilinear and trilinear varieties for two and three perspective views. As real motion sequences contain motions of different types, we cast motion segmentation as a problem of clustering manifolds of different types. Rather than explicitly modeling each manifold as a linear, bilinear or multilinear variety, we use nonlinear dimensionality reduction to learn a low-dimensional representation of the union of all manifolds. We show that for a union of separated manifolds, the LLE algorithm computes a matrix whose null space contains vectors giving the segmentation of the data. An analysis of the variance of these vectors allows us to distinguish them from other vectors in the null space. This leads to a new algorithm for clustering both linear and nonlinear manifolds. Although this algorithm is theoretically designed for separated manifolds, our experiments demonstrate its performance on real data where this assumption does not hold. We test our algorithm on the Hopkins 155 *motion segmentation database and achieve an average classification error of* 4.8%*, which compares favorably against state-of-the art multiframe motion segmentation methods.*

1. Introduction

Recent years have witnessed an increasing interest on the understanding of *dynamic scenes* in which both the camera and multiple objects move. This is a non-trivial problem as it requires the simultaneous estimation of an unknown number of motion models, without the knowledge of which measurements correspond to which model.

In the case of affine cameras, the trajectories associated with each rigidly moving object live in a linear subspace of dimension four or less [1, 15]. Thus, motion segmentation is equivalent to clustering linear subspaces. Earlier work [1, 2] was based on thresholding the leading singular vector of the matrix of point trajectories or the entries of the so-called *shape interaction* matrix. However, this thresholding process is very sensitive to noise [3, 6]. Another disadvantage of these methods is that they require the motion subspaces to be linearly independent [6], thus they are not provably correct for most practical motion sequences which usually exhibit partially dependent motions. This has led to the development of several algorithms for dealing with partially dependent motions, including statistical methods [12], spectral methods [23] and algebraic methods [17]. In the case of perspective cameras, the situation is more complicated as the trajectories associated with different moving objects live in different multilinear varieties. Prior work has been limited to algebraic methods for two [20, 21] and three [4] views and statistical methods for multiple views [10].

In this paper, we seek an alternative approach that deals automatically with motions of different types. Rather than explicitly modeling each manifold as a linear, bilinear or multilinear variety, we use nonlinear dimensionality reduction (NLDR) to learn a low-dimensional representation of the union of all manifolds. NLDR refers to the problem of finding a low-dimensional representation for a set of points lying in a nonlinear manifold embedded in a high dimensional space. A huge family of algorithms computes a low-dimensional representation from the top or bottom eigenvectors of a matrix M constructed from the local geometry of the manifold. Algorithms in this family include ISOMAP [13] and locally linear embedding (LLE) [9].

Although the goals of dimensionality reduction, classification and segmentation have always been intertwined with each other, considerably less work has been done on extending NLDR techniques for the purpose of clustering data living on different manifolds. In the case of linear manifolds, there are many existing subspace clustering methods including K-subspaces [5], Generalized Principal Component Analysis (GPCA) [19], and Mixtures of Probabilistic PCA (MPPCA) [14]. However, all subspace clustering methods are formulated only for mixtures of linear manifolds and do not work in the presence of nonlinear manifolds. Existing NLDR techniques for manifold clustering include [11] and [8]. [11] is an EM-like extension of ISOMAP for clustering multiple nonlinear manifolds. This method is very sensitive to good initialization and uses heuristics in the E-step to assign points to manifolds. The work of [8] applies LLE to a manifold with m connected components. It shows that m eigenvalues of the matrix M are zero and that the clustering of the data can be obtained from the corresponding eigenvectors. However, this LLE clustering algorithm suffers from degeneracies in the presence of subspaces, as we will see in §3.3.

Paper contributions. In this paper, we study the problem of segmenting multiple motions of different types. We cast this problem as one of simultaneous nonlinear dimensionality reduction and clustering of both linear and nonlinear manifolds. In particular, we study the mathematical properties of the LLE algorithm for manifold clustering and demonstrate that it becomes degenerate when the data points are drawn from a union of linear manifolds, *i.e.* subspaces. In spite of these degeneracies, we show that one can still obtain the membership of the data as well as a low-dimensional representation for each one of the manifolds.

2. Multiframe Motion Segmentation Problem

2.1. Segmentation from Multiple Affine Views

Let $\{\mathbf{x}_{fp} \in \mathbb{R}^2\}_{p=1,\dots,n}^{f=1,\dots,F}$ be the projections of n 3-D points $\{\mathbf{X}_p \in \mathbb{P}^3\}_{p=1}^n$ lying on a rigidly moving object onto F frames of a rigidly moving camera. Under the affine projection model, the images satisfy the equation

$$\mathbf{x}_{fp} = A_f \mathbf{X}_p, \tag{1}$$

where $A_f \in \mathbb{R}^{2\times 4}$ is the *affine camera matrix* at frame f.

Let $W_1 \in \mathbb{R}^{2F\times n}$ be the matrix whose n columns are the image point trajectories $\{\mathbf{x}_{fp}\}_{p=1}^n$. It follows from (1) that W_1 can be decomposed into a *motion matrix* $M_1 \in \mathbb{R}^{2F\times 4}$ and a *structure matrix* $S_1 \in \mathbb{R}^{n\times 4}$ as

$$\begin{aligned} W_1 &= M_1 S_1^\top \\ \begin{bmatrix} \mathbf{x}_{11} \cdots \mathbf{x}_{1n} \\ \vdots \quad\quad \vdots \\ \mathbf{x}_{F1} \cdots \mathbf{x}_{Fn} \end{bmatrix}_{2F\times n} &= \begin{bmatrix} A_1 \\ \vdots \\ A_F \end{bmatrix}_{2F\times 4} \begin{bmatrix} \mathbf{X}_1 \cdots \mathbf{X}_P \end{bmatrix}_{4\times n}, \end{aligned} \tag{2}$$

hence $\operatorname{rank}(W_1) \le 4$. Since the affine camera matrix A_f is full rank, we also have that $\operatorname{rank}(W_1) \ge \operatorname{rank}(A_f) = 2$. Therefore, under the affine projection model, the 2-D trajectories of a set of 3-D points seen by a rigidly moving camera (the columns of W_1) live in a subspace of $\mathbb{R}^{2F}$ of dimension $d_1 = \operatorname{rank}(W_1) = 2$, 3 or 4 [15].

Assume now that the n trajectories $\{\mathbf{x}_{fp}\}_{p=1}^n$ correspond to m objects undergoing m rigid-body motions relative to a moving camera. The 3-D motion segmentation problem is the task of clustering these n trajectories according to the m moving objects and is equivalent to clustering a set of points into m subspaces of $\mathbb{R}^{2F}$ of unknown dimensions $d_j \in \{2,3,4\}$ for $j = 1,\dots,m$.

2.2. Segmentation from Multiple Perspective Views

Under the perspective projection model, point trajectories associated with m moving bodies live in m bilinear varieties for two views and trilinear varieties for three views [7]. Thus, 3-D motion segmentation is equivalent to clustering data lying in m bilinear or trilinear varieties. In the general case of F views, the relationships among point correspondences are multilinear on the point correspondences. However, these multilinear constraints are algebraically dependent on those among two and three views [7]. Furthermore, it is well known that, except for degenerate cases, the multilinear constraints are algebraically dependent on the bilinear constraints. It is shown in [18] that among all bilinear constraints between all pairs of views, only $2F-3$ are algebraically independent. Therefore, the point correspondences $[\mathbf{x}_{1p}, \mathbf{x}_{2p}, \dots, \mathbf{x}_{Fp}]^\top \in \mathbb{R}^{2F}$ live in a manifold of dimension $2F-(2F-3) = 3$. 3-D motion segmentation is then equivalent to clustering these m manifolds.

2.3. Segmentation of Motions of Different Types

In general, scenes will have multiple moving objects occupying a small area of the image and thus their motion can be well approximated by the affine model. The background points, on the other hand, describe the motion of the camera, which usually has significant perspective effects due to depth variations, forward motions, etc., thereby requiring the perspective model. Therefore, real motion sequences contain motions of different types, and there is a need for developing methods that deal automatically with subspaces of dimension $2,3$, or 4 or multilinear manifolds of dimension 3. We develop such a method in the next section.

3. Locally Linear Manifold Clustering: LLMC

This section presents our algorithm for simultaneous nonlinear dimensionality reduction and manifold clustering.

3.1. Locally Linear Embedding

Let $X = \{\mathbf{x}_i \in \mathbb{R}^D\}_{i=1}^n$ be a set of n points sampled from a low-dimensional submanifold of $\mathbb{R}^D$. We assume that the n points are k-connected, *i.e.* for any two points $\mathbf{z}_1, \mathbf{z}_2 \in X$ there is an ordered sequence of points in X having $\mathbf{z}_1$ and $\mathbf{z}_2$ as endpoints, such that any two consecutive points in the sequence have at least one k nearest neighbor in common.

Given a k-connected set of points, the goal of dimensionality reduction is to find a set of n vectors $\{\mathbf{y}_i \in \mathbb{R}^d\}_{i=1}^n$, where $d \ll D$, such that nearby points remain close and distant points remain far. Locally Linear Embedding (LLE) [9] is a simple and elegant algorithm for nonlinear dimensionality reduction. LLE exploits the fact that the local neighborhood of a point on the manifold can be well approximated by the affine subspace spanned by its k nearest neighbors. The LLE algorithm can be summarized as follows:

1. *Nearest neighbors search:* For each data point $\mathbf{x}_i \in \mathbb{R}^D$, find its k nearest neighbors (kNN) $\{\mathbf{x}_j\}$.
2. *Least squares fit*: Find a matrix of weights $W \in \mathbb{R}^{n\times n}$ whose entries W_{ij} minimize the reconstruction error
$$\varepsilon(W) = \sum_{i=1}^{n} \|\mathbf{x}_i - \sum_{j=1}^{n} W_{ij}\mathbf{x}_j\|^2 \quad (3)$$
subject to the constraints (i) $W_{ij} = 0$ if $\mathbf{x}_j$ is not a k-nearest neighbor of $\mathbf{x}_i$ and (ii) $\sum_{j=1}^{n} W_{ij} = 1$.
3. *Sparse eigenvalue problem*: Find a matrix $Y \in \mathbb{R}^{n\times d}$ whose rows $\mathbf{y}_i \in \mathbb{R}^d$ minimize the error
$$\phi(Y) = \sum_{i=1}^{n} \|\mathbf{y}_i - \sum_{j=1}^{n} W_{ij}\mathbf{y}_j\|^2 \quad (4)$$
subject to the constraints (i) $\sum_{i=1}^{n} \mathbf{y}_i = 0$ (centered at the origin) and (ii) $\frac{1}{n}\sum_{i=1}^{n} \mathbf{y}_i\mathbf{y}_i^\top = I$ (unit covariance). The solution to this problem is given by the matrix Y whose columns are the d eigenvectors of the symmetric semi-positive definite matrix $M = (I-W)^\top(I-W)$ associated with its second to $(d+1)$-th smallest eigenvalues. The first eigenvector of M is discarded, because it is the vector of all ones, $\mathbf{1} \in \mathbb{R}^n$, with 0 as its eigenvalue. This is because $\sum_{j=1}^{n} W_{ij} = 1$, hence $W\mathbf{1} = \mathbf{1}$.

3.2. LLMC for a disconnected union of k-connected nonlinear manifolds

In this subsection, we propose an extension of the LLE algorithm for clustering a union of m k-connected manifolds under the assumption that **no kNN of a data point in one manifold lies in a different manifold**. In principle, this could be considered as a strong assumption, because it would allow one to cluster the different groups by simply searching for the connected components of the graph [9]. However, we will show in §4 that with real data, where the assumption of separated manifolds is violated, clustering the data points by simply looking at the connected components of the graph does not yield comparable performance with respect to our proposed method. In addition, when searching for the connected components, it is not clear how to determine whether an edge should be considered as weak or strong without imposing different heuristics for each different dataset, hence requiring supervision of the algorithm.

The following proposition shows how to apply LLE for clustering a union of m k-connected nonlinear manifolds. Contrary to intuition, the case of nonlinear manifolds is simpler than the case of linear subspaces, as we will see in §3.3.

Proposition 1 *[8] Let $\{\mathbf{x}_i\}_{i=1}^n$ be a set of points drawn from a disconnected union of m k-connected nonlinear manifolds of dimension $d < k-1$. Then, there exist m vectors $\{\mathbf{v}_j\}_{j=1}^m$ in the null space of M such that $\mathbf{v}_j$ corresponds to the jth group of points,* i.e. $\mathbf{v}_{ij} = 1$ *if the ith data point is in the jth group, and $\mathbf{v}_{ij} = 0$ otherwise.*

Proof. If the data can be partitioned into m k-connected groups, then the matrix W is block-diagonal with m blocks. This is because if points $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to different groups, then they cannot be kNN of each other, hence $W_{ij} = 0$. We can write $W = \text{diag}(W_j)$, where $W_j \in \mathbb{R}^{n_j\times n_j}$ is the matrix for the j-th group, which contains n_j points. Since the matrix M is given by $M = (I-W)^\top(I-W)$, it is clear that the matrix M is also block diagonal, and we can write it as $M = \text{diag}(M_j)$, where $M_j \in \mathbb{R}^{n_j\times n_j}$ is the matrix for the j-th group. From the properties of the LLE algorithm, we know that each one of the m blocks of M, has the vector $\mathbf{1} \in \mathbb{R}^{n_j}$ in its null space. Therefore, there are m vectors $\{\mathbf{v}_j\}$ in $\ker(M)$, with each $\mathbf{v}_j$ taking the values 1 and 0, indicating the group membership, as claimed. ■

Notice that when computing a basis $B \in \mathbb{R}^{n\times m}$ for $\ker(M)$, we do not necessarily obtain the set of membership vectors, but rather linear combinations of them, including the vector $\mathbf{1}$. Nevertheless, these linear combinations still contain the segmentation of the data, so we can cluster the data into m groups by applying K-means to the rows of B.

3.3. LLMC for a disconnected union of k-connected linear manifolds

LLE for a single subspace. For the sake of simplicity, let us first consider the application of LLE to a single k-connected linear manifold. Intuitively we would expect that if LLE is applied to a dataset that is already a subspace of dimension d, the output representation should again be a subspace of the same dimension. Proposition 2 below shows that when d is known, the low-dimensional representation is indeed a subspace of dimension d, which is contained in the null space of the matrix M representing the local geometry of the manifold. As the vector of all ones $\mathbf{1}$ is also in $\ker(M)$, this could cause some degeneracies when applying LLE to linear subspaces. An extension of this result to a locally flat manifold can be found in [8].

Proposition 2 *Assume that the data points $\mathbf{x}_i \in \mathbb{R}^D$ lie in a subspace of $\mathbb{R}^D$ of dimension $d < k-1$. Then the dimension of the null space of M is at least $d+1$.*

Proof. Since the data lie in a subspace of $\mathbb{R}^D$ of dimension $d < k-1$, each point $\{\mathbf{x}_i\}$ can be reconstructed with zero error in (3), *i.e.* for all $i = 1,\dots,n$, we have $\mathbf{x}_i = \sum_{j=1}^{n} W_{ij}\mathbf{x}_j$. If we let $X \in \mathbb{R}^{n\times D}$ be the matrix whose rows are the data points, then we have that $WX = X$, hence $MX = 0$. In other words, the vector of each one of the coordinates of the given data set is in the null space of M. As $\text{rank}(X) = d$, the null space of M is at least d-dimensional. On the other hand, since the data points live in a subspace of dimension d, there exist a matrix $B \in \mathbb{R}^{D\times d}$ and vectors $\{\mathbf{y}_i\}$ such that $\mathbf{x}_i = B\mathbf{y}_i + \mathbf{m}$, so that $\sum_{i=1}^{n} \mathbf{y}_i = 0$, where $\mathbf{m} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \in \mathbb{R}^D$ is the mean of the data. This implies that $\mathbf{y}_i = \sum_{j=1}^{n} W_{ij}\mathbf{y}_j$; hence $MY = 0$, where $Y \in \mathbb{R}^{n\times d}$ is a matrix whose rows are the $\{\mathbf{y}_i\}$ vectors. Now, by construction, the vector of all ones $\mathbf{1}$ is also in $\ker(M)$, because $\sum_{j=1}^{n} W_{ij} = 1$. Since in addition $\sum_{i=1}^{n} \mathbf{y}_i = 0$, we have that $\mathbf{1}^\top Y = \mathbf{0}^\top$, hence the vector $\mathbf{1}$ is linearly independent from the columns of Y. Therefore, the null space of M is at least $(d+1)$-dimensional. ∎

From Proposition 2, we see that if we apply LLE to data lying in a subspace of dimension d and choose the second to $(d+1)$-th smallest eigenvectors of M for dimensionality reduction, we might not get the correct subspace reconstruction. This is because the the embedding eigenvectors may be mixed with the vector $\mathbf{1}$, which is also a null vector of M.

LLMC for multiple subspaces. Consider now the problem of clustering data points $\{\mathbf{x}_i\}_{i=1}^n$ drawn from a union of m k-connected subspaces of $\mathbb{R}^D$ with dimensions $\{d_j\}_{j=1}^m$. From Propositions 1 and 2, we know that there are two types of vectors in the null space of M: the embedding vectors coming from the matrix of coordinates and the membership vectors coming from each one of the m connected components. However, it is unclear if these vectors are linearly independent, and if one can recover the segmentation of the data and a nonlinear embedding for each group from $\ker(M)$.

The following propositions address these issues in detail. Proposition 3 derives the dimension and a basis for the null space of M. Proposition 4 shows that the membership eigenvectors have smaller variance than the embedding eigenvectors. Proposition 5 shows that one can obtain the membership eigenvectors by solving a generalized eigenvalue problem.

Proposition 3 *Let $\{\mathbf{x}_i\}_{i=1}^n$ be a set of points drawn from m k-connected subspaces of dimension $d_j < k-1$. The null space of M is of dimension at least $m + \sum_{j=1}^m d_j$ and contains orthonormal zero-padded vectors formed from the individual embedding and membership vectors.*

Proof. From the proof of Proposition 1, we know that M is block diagonal and can be written as $M = \operatorname{diag}(M_j)$, where $M_j \in \mathbb{R}^{n_j \times n_j}$ is the matrix for the j-th group, and n_j is the number of points in the j-th group. From the proof of Proposition 2 we also know that the matrix M_j has d_j+1 vectors in the null space: the vector of all ones and the d_j linearly independent columns of the matrix of coordinates $Y_j \in \mathbb{R}^{n_j \times d_j}$. That is $M_j\,[Y_j\,\mathbf{1}] = 0$. Therefore, the matrix $Y = \operatorname{diag}\left([Y_{j(n_j\times d_j)}\,\mathbf{1}_{(n_j\times 1)}]\right) \in \mathbb{R}^{n\times(\sum_{j=1}^m d_j + m)}$ is such that $MY = 0$. Furthermore, as Y is block diagonal and $\operatorname{rank}\left([Y_j\,\mathbf{1}]\right) = d_j + 1$, we have that Y is of rank $\sum_{j=1}^m d_j + m$, and so the dimension of $\ker(M)$ is at least $\sum_{j=1}^m d_j + m$. Also, the embedding vector of the j-th group $\mathbf{e}_j$ is orthogonal to its membership vector $\mathbf{v}_j$, and because $\mathbf{e}_j$ and $\mathbf{v}_j$ are zero-padded, they are always orthogonal to $\mathbf{e}_i$ and $\mathbf{v}_i$ for $i \neq j$. In addition, one can choose the embedding vectors $\mathbf{e}_j$ to be orthogonal to each other, because the matrix M is symmetric. Therefore, from now on we will assume that the vectors $\{\mathbf{v}_1, \mathbf{e}_1, \ldots, \mathbf{v}_m, \mathbf{e}_m\}$ are orthonormal. ∎

From the proof of Proposition 3, we know that $\ker(M)$ contains the embedding vectors $\mathbf{e}_j = [\mathbf{0}, Y_{j(d_j\times n_j)}^\top, \mathbf{0}]^\top \in \mathbb{R}^{n\times d_j}$ and the membership vectors $\mathbf{v}_j = [\mathbf{0}, \mathbf{1}_{(1\times n_j)}, \mathbf{0}]^\top \in \mathbb{R}^{n\times 1}$. Therefore, we cannot directly obtain the segmentation of the data or an embedding for each one of the subspaces from $\ker(M)$, because an arbitrary vector in $\ker(M)$ is a linear combination of both membership and embedding vectors.

In order to distinguish between membership and embedding vectors, we look at the variance of the eigenvectors of M and show that the segmentation eigenvectors are those with smaller variance. To this end, recall that for any $n\times 1$ vector $\mathbf{u}$, the mean of all its entries is $\bar{\mathbf{u}} = \frac{1}{n}\mathbf{1}_{1\times n}\mathbf{u}$ and the variance of all its entries is

$$\mathbf{var}(\mathbf{u}) = \frac{1}{n}\sum_{k=1}^{n}(\mathbf{u}_k - \bar{\mathbf{u}})^2 = \frac{1}{n}\mathbf{u}^\top (I-J)\mathbf{u}, \qquad (5)$$

where $J = \frac{1}{n}\mathbf{1}_{n\times 1}\mathbf{1}_{1\times n}^\top$. We then have the following result.

Proposition 4 *Let $\{\mathbf{v}_j\}$ and $\{\mathbf{e}_j\}$ be the (unit) membership and embedding vectors in the null space of M respectively.*

1. *For all i and j we have $\mathbf{var}(\mathbf{v}_i) < \mathbf{var}(\mathbf{e}_j)$.*
2. *For all α_i and β_j such that $\|\sum_i \alpha_i \mathbf{v}_i\| = \|\sum_j \beta_j \mathbf{e}_j\| = 1$, we have $\mathbf{var}(\sum_i \alpha_i \mathbf{v}_i) < \mathbf{var}(\sum_j \beta_j \mathbf{e}_j)$.*
3. *If $\|\sum_i \alpha_i \mathbf{v}_i + \sum_j \beta_j \mathbf{e}_j\| = 1$, then $\mathbf{var}(\sum_i \alpha_i \mathbf{v}_i + \sum_j \beta_j \mathbf{e}_j)$ is minimized when $\beta_j = 0, \forall j$.*

Proof.

1. By definition of an embedding vector, we have $\bar{\mathbf{e}}_j = 0$. Also, since the eigenvectors of M are assumed to be of unit norm, *i.e.* $\|\mathbf{e}_j\| = \|\mathbf{v}_j\| = 1$, then each nonzero entry of $\mathbf{v}_j$ must be equal to $\frac{1}{\sqrt{n_j}}$, so that $\bar{\mathbf{v}}_j = \frac{\sqrt{n_j}}{n}$. Therefore, $\forall i, j \quad \mathbf{var}(\mathbf{v}_i) = \frac{1}{n} - \frac{n_i}{n^2} < \frac{1}{n} = \mathbf{var}(\mathbf{e}_j)$.
2. Since $\{\mathbf{v}_j\}$ and $\{\mathbf{e}_j\}$ are orthonormal vectors, then $\|\sum \alpha_i \mathbf{v}_i\| = \|\sum \beta_i \mathbf{e}_i\| = 1$ implies $\sum_i \alpha_i^2 = \sum_j \beta_j^2 = 1$. Thus, $\mathbf{var}(\sum_i \alpha_i \mathbf{v}_i) = \sum_i \alpha_i^2 \mathbf{var}(\mathbf{v}_i) = \frac{1}{n} - \sum_i \frac{\alpha_i^2 n_i}{n^2} < \frac{1}{n} = \sum_j \beta_j^2 \mathbf{var}(\mathbf{e}_j) = \mathbf{var}(\sum_j \beta_j \mathbf{e}_j)$.
3. $\|\sum_i \alpha_i \mathbf{v}_i + \sum_j \beta_j \mathbf{e}_j\| = 1$ implies $\sum_i \alpha_i^2 + \sum_j \beta_j^2 = 1$. Thus $\mathbf{var}(\sum_i \alpha_i \mathbf{v}_i + \sum_j \beta_j \mathbf{e}_j) = \frac{1}{n} - \sum_i \frac{\alpha_i^2 n_i}{n^2}$. One can show that this quantity achieves its minimum value when $\sum_i \alpha_i^2 = 1$, so $\sum_j \beta_j^2 = 0$ and $\beta_j = 0, \forall j$. ∎

If we had the ability of computing each one of the vectors $\mathbf{e}_i$ and $\mathbf{v}_i$ from $\ker(M)$, then we could find the membership eigenvectors as those with smaller variance. However, as we alluded to earlier, we cannot directly compute such membership and embedding eigenvectors, but only a basis $B \in \mathbb{R}^{n\times(m+\sum d_i)}$ for $\ker(M)$. Given such a basis, both segmentation and embedding eigenvectors can be expressed as $B\alpha$, where $\|B\alpha\| = 1$ for $\alpha \in \mathbb{R}^{m+\sum d_i}$. Since the variance of $B\alpha$ is $\alpha^\top B^\top (I-J)B\alpha/n$, we can search for the membership eigenvectors by choosing a vector α that minimizes this variance. Therefore, we have the following result.

Proposition 5 *The membership eigenvectors* $\{\mathbf{v}_i\}_{i=1}^m$ *can be computed as* $\mathbf{v}_i = BQ^{-\frac{1}{2}}\beta_i$*, where* β_i *are the eigenvectors of* $Q^{-\frac{1}{2}}B^\top(I-J)BQ^{-\frac{1}{2}}$ *associated with its smallest* m *eigenvalues and* $Q = B^\top B$.

Proof. Let $\lambda_{\min}$ be the solution to

$$\min_{\|B\alpha\|=1} \alpha^\top B^\top (I-J)B\alpha = \min_{\alpha\neq\mathbf{0}} \frac{\alpha^\top B^\top (I-J)B\alpha}{\alpha^\top B^\top B\alpha}.$$

This is a generalized eigenvalue problem of the form

$$B^\top(I-J)B\alpha = \lambda_{\min}B^\top B\alpha = \lambda_{\min}Q\alpha,$$

with $B^\top B = Q$ being symmetric positive definite. Thus, $Q^{-\frac{1}{2}}B^\top(I-J)BQ^{-\frac{1}{2}}\beta = \lambda_{\min}\beta$ and $\beta = Q^{\frac{1}{2}}\alpha$. ∎

Thanks to Proposition 5, we can cluster the data by applying K-means to the rows of $[\mathbf{v}_1,\ldots,\mathbf{v}_m] \in \mathbb{R}^{n\times m}$.

3.4. LLMC for a union of k-connected linear and nonlinear manifolds

Let $\{\mathbf{x}_i\}_{i=1}^n$ be a set of points drawn from m k-connected manifolds, of which m_1 are subspaces of dimensions d_j, $j = 1,\cdots,m_1$. The dimension of the null space of M is at least $m + \sum_{j=1}^{m_1} d_j$, with $m_1 < m$. To cluster both linear and nonlinear manifolds, we proceed as discussed in Proposition 5. This gives the following algorithm for manifold clustering.

Locally Linear Manifold Clustering Algorithm

Let $X = \{\mathbf{x}_i \in \mathbb{R}^D\}_{i=1}^n$ be a set of n data points sampled from m k-connected manifolds of $\mathbb{R}^D$.

1. Apply the LLE algorithm to the entire dataset to obtain the matrix M.
2. Compute a basis B for the null space of M.
3. Compute the matrix $Q = B^\top B$ and the m eigenvectors of $Q^{-\frac{1}{2}}B^\top(I-J)BQ^{-\frac{1}{2}}$, $\{\beta_i\}_{i=1}^m$, whose corresponding eigenvalues are less than $\frac{1}{n}$.
4. Apply K-means to the rows of the matrix of membership vectors $S = [BQ^{-\frac{1}{2}}\beta_1,\cdots,BQ^{-\frac{1}{2}}\beta_m]$ to cluster the original data points into m different groups.
5. Apply LLE to each group to obtain a low-dimensional embedding for each manifold.

4. Experiments

In this section, we test our 3-D motion segmentation algorithm on the Hopkins 155 motion segmentation database [16] which is available at `http://www.vision.jhu.edu`. The database consists of 155 sequences of both indoor and outdoor scenes containing two or three motions. The database is divided into three main groups, *checkerboard, traffic* and *articulated* sequences. We compare the results of LLMC with those of the following clustering algorithms.

1. Multi-Stage Learning (MSL) [12]: this statistical approach solves a series of optimization problems iteratively using the Expectation Maximization algorithm.
2. GPCA [17]: this algebraic approach is based on fitting a union of m subspaces with a polynomial of degree m. The gradient of this polynomial at a point gives a vector normal to the subspace containing that point. The data are segmented by applying spectral clustering to a similarity built from the angles between these normals.
3. Local Subspace Affinity (LSA) [22]: this method projects the data onto a low-dimensional subspace and then fits a subspace to each projected point and its k nearest neighbors. The data are segmented by applying spectral clustering to a similarity built from the principal angles between these locally computed subspaces.
4. Connected Components Search (CCS) [9]: this method searches for the connected components of the graph using Tarjan's algorithm on the neighborhood graph obtained in the first step of LLE.

Before applying LLMC or LSA, we project the trajectories onto a subspace of dimension 5 or $4m$, where m is the number of motions, as suggested in [16]. We refer to these variants as LLMC5, LLMC4m, LSA5 and LSA4m.

Table 1 contains the average and median classification errors given by each algorithm on the 155 sequences, and Figure 1 shows histograms of these errors. Notice that MSL gives nearly perfect segmentation for the majority of the sequences, but it occasionally gives large errors when it converges to a local minimum. LLMC, on the other hand, gives a small error for the majority of the sequences, but also gives large errors for a few sequences. For three motions, GPCA, LSA5 and CCS do not perform as well as LLMC, LSA4m and MSL, as shown in Figure 1. LLMC yields exceptional results on the traffic sequences, and performs reasonably well on the checkerboard sequences. Finally, even though LLMC assumes that the manifolds are separated, it substantially outperforms CCS. This suggests that LLMC is much more effective and sophisticated than CCS for real data with noise. On average, our algorithm gives a classification error of 4.8% for LLMC5 and 5.93% for LLMC4m for all sequences, while MSL gives 5.06%, GPCA 9.18%, LSA5 11.82%, LSA4m 4.87%, and CCS 15.37%.

5. Conclusions

We have presented a novel algorithm for segmenting motions of different types. Unlike existing approaches which assume in-depth knowledge about the type of motions in the scene, LLMC is an unsupervised method based on simultaneous dimensionality reduction and clustering for data lying in m separated k-connected manifolds. Experiments on 155 motion sequences showed that LLMC matches the performance of state-of-the-art motion segmentation algorithms.

Table 1. Misclassification rates (%) for motion database

	LLMC 5	LLMC 4*m*	MSL	GPCA	LSA 5	LSA 4*m*	CCS
Entire dataset with 2 and 3 motions							
Mean	4.80	5.93	5.06	10.02	11.82	4.87	15.37
Median	0.00	0.63	0.00	2.39	4.00	0.90	4.47
Number of motions=2							
Checkerboard							
Mean	4.37	4.65	4.46	6.09	8.84	2.57	16.37
Median	0.00	0.11	0.00	1.03	3.43	0.27	10.62
Traffic							
Mean	0.84	3.65	2.23	1.41	2.15	5.43	5.27
Median	0.00	0.33	0.00	0.00	1.00	1.48	0.00
Articulated							
Mean	6.16	5.23	7.23	2.88	4.66	4.10	17.58
Median	1.37	1.30	0.00	0.00	1.28	1.22	7.07
All							
Mean	3.62	4.44	4.14	4.59	6.73	3.45	12.16
Median	0.00	0.24	0.00	0.38	1.99	0.59	0.00
Number of motions=3							
Checkerboard							
Mean	10.70	12.01	10.38	31.95	30.37	5.80	28.63
Median	9.21	9.22	4.61	32.93	31.98	1.77	33.21
Traffic							
Mean	2.91	7.79	1.80	19.83	27.02	25.07	3.02
Median	0.00	5.47	0.00	19.55	34.01	23.79	0.18
Articulated							
Mean	5.60	9.38	2.71	16.85	23.11	7.25	44.89
Median	5.60	9.38	2.71	16.85	23.11	7.25	44.89
All							
Mean	8.85	11.02	8.23	28.66	29.28	9.73	26.18
Median	3.19	6.81	1.76	28.26	31.63	2.33	31.74

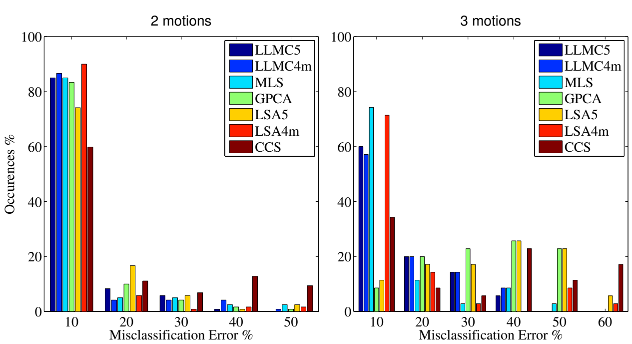

Figure 1. Misclassification rates of two and three motions

Acknowledgements

Work supported by startup funds from Johns Hopkins University, and by grants NSF CAREER IIS-04-47739, NSF EHS-05-09101 and ONR N00014-05-1083.

References

[1] T.E. Boult and L.G. Brown. Factorization-based segmentation of motions. *Workshop on Motion Understanding*, pages 179-186, 1991.

[2] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[3] C. W. Gear. Multibody grouping from motion images. *International Journal of Computer Vision*, 29(2):133–150, 1998.

[4] R. Hartley and R. Vidal. The multibody trifocal tensor: Motion segmentation from 3 perspective views. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 769–775, 2004.

[5] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman. Clustering apperances of objects under varying illumination conditions. *IEEE Conf. on Computer Vision and Pattern Recogn.*, pages 11–18, 2003.

[6] K. Kanatani. Motion segmentation by subspace separation and model selection. *IEEE International Conference on Computer Vision*, volume 2, pages 586–591, 2001.

[7] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry. *An Invitation to 3D Vision: From Images to Geometric Models*. Springer Verlag, 2003.

[8] M. Polito and P. Perona. Grouping and dimensionality reduction by locally linear embedding. *Neural Inf. Processing Systems*, 2002.

[9] S. Roweis and L. Saul. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003.

[10] K. Schindler, J. U, and H. Wang. Perspective n -view multibody structure-and-motion through model selection. *European Conference on Ccomputer Vision*, pages 606–619, 2006.

[11] R. Souvenir and R. Pless. Manifold clustering. *IEEE International Conference on Computer Vision*, volume I, pages 648–653, 2005.

[12] Y. Sugaya and K. Kanatani. Geometric structure of degeneracy for multi-body motion segmentation. *Workshop on Statistical Methods in Video Processing*, 2004.

[13] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[14] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999.

[15] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[16] R. Tron and R. Vidal. A benchmark for the comparsion of 3-D motion segmentation algorithms. *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[17] R. Vidal and R. Hartley. Motion segmentation with missing data by PowerFactorization and Generalized PCA. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 310–316, 2004.

[18] R. Vidal, Y. Ma, S. Hsu, and S. Sastry. Optimal motion estimation from the multiview normalized epipolar constraint. *IEEE International Conference on Computer Vision*, pages 34–41, Vancouver, Canada, 2001.

[19] 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.

[20] R. Vidal, Y. Ma, S. Soatto, and S. Sastry. Two-view multibody structure from motion. *International Journal of Computer Vision*, 68(1):7–25, 2006.

[21] L. Wolf and A. Shashua. Two-body segmentation from two perspective views. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 263–270, 2001.

[22] J. Yan and M. Pollefeys. A factorization approach to articulated motion recovery. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–821, 2005.

[23] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. *European Conference on Computer Vision*, pages 94–106, 2006.