

# Sparse Optimization for Motion Segmentation

Michael Ying Yang<sup>(✉)</sup>, Sitong Feng, and Bodo Rosenhahn

Institute for Information Processing (TNT), Leibniz University Hannover,  
Hannover, Germany  
yang@tnt.uni-hannover.de

**Abstract.** In this paper, we propose a new framework for segmenting feature-based multiple moving objects with subspace models in affine views. Since the feature data is high-dimensional and complex in the real video sequences, most traditional approaches for motion segmentation use the conventional PCA to obtain a low-dimensional representation, while our proposed framework applies the sparse PCA (SPCA) to obtain a projected subspace, which is a low-dimensional global subspace on a Stiefel manifold with sparse entries. Then, the local subspace separation is achieved via automatically selecting the sparse nearest neighbours. By combining two sparse techniques, the proposed framework segments different motions through a simple spectral clustering on an affinity matrix built with the principal angles. To the best of our knowledge, our framework is the first one to apply the sparse optimization for optimizing the global and local subspace simultaneously. We test our method extensively and compare its performance to several state-of-art motion segmentation methods with experiments on the Hopkins 155 dataset. Our results are comparable with these results, and in many cases exceed them both in terms of segmentation accuracy and computational speed.

## 1 Introduction

Motion segmentation aims to decompose a video sequence into different moving objects that move throughout the sequence. In the recent years, the tracked features based motion segmentation problem has motivated amount of people to find a fast and high accuracy method. Particularly, different with the traditional motion segmentation method based on the pixel-wise model such as [1, 2], which is focused on segmenting the foreground moving objects from their background in an unannotated video, the segmentation of only the few number of tracked features on each moving object can not only solve the occlusions in the video, but also save computation time w.r.t. the pixel-wise methods. Motion segmentation approaches which based on the tracked features of a moving body focus on clustering the sparse or dense feature points into different regions and each region represents a moving object in the sequence.

The goal of a general feature-based motion segmentation is to cluster the union of different point trajectories with different labels and different labels represent the different motions, as shown in Fig. 1. We address the feature trajectories segmentation as a subspace clustering problem under the affine camera

model. Under the model of subspace, the trajectories which can be extracted during a preprocessing step, such as KLT [3], SIFT [4] or SURF [5], are embedded in a union of local subspaces and each trajectory is represented with a local subspace. Then, the problem of segmenting different feature trajectories changes to cluster the union of different local subspaces.



**Fig. 1.** Examples of our segmentation result on the real sequences *cars9* and *kananchi1* from the Hopkins155 dataset. (The first row: *from left to right* are the results for frame 1, 40 and 50; the second row: *from left to right* are the results for frame 1, 10 and 20.

**Contributions.** Our main contribution is an efficient framework for motion segmentation which based on subspace clustering with sparse optimization. We combine two state-of-art sparse representations to optimize both the global and local estimation. *Sparse Principle Components Analysis*(SPCA) is applied for the global optimization, in the same time, we seek a sparse representation for the closest neighbours for the local subspace separation with computing principal angles. As illustrated in Fig. 1, our method can clearly label the moving objects that tracked throughout a video. To the best of our knowledge, our framework is the first one to apply the sparse optimization for optimizing the global and local subspace simultaneously.

The following sections are organized as follows. The related works are discussed in Sect. 2. Section 3 introduces the subspace models for motion segmentation. Our proposed approach described in detail in Sect. 4. In Sect. 5, experimental results are presented. Finally, this work is concluded and future work is discussed in Sect. 6.

## 2 Related Work

There are a large amount of works on motion segmentation. In general, the methods for motion segmentation can be divided into two classes: affinity-based and

subspace-based methods. The affinity-based approach [6] is based on computing the affinity between a pair of trajectories. Our approach is concentrated on the subspace methods, which focuses on segmentation of the different motions with finding the membership from a union of subspaces. The subspace-based methods can be divided into: iterative methods, algebraic solution, compressive sensing and subspace estimation. One of the iterative methods is RANSAC algorithm [7], which can deal with outliers and noise. But the number of subspaces needed to be known as prior knowledge, and they require a good initial estimation and parameter selection. The most popular method based on the algebraic solution is the *Generalized Principal Component Analysis* (GPCA) [8]. While GPCA gives a good performance for the subspaces with different dimensions, but GPCA is not robust to the noise and outliers. Agglomerate Lossy Compression (ALC) [9] is a method using the compressive sensing on subspace model. The ALC method is robust to noise and outliers without knowing the subspace dimension and the number of the subspace, but it is highly time-consuming. The other application of compressive sensing on subspace models is based on the sparse representation. One of the most popular methods is the Sparse Subspace Clustering (SSC) [10]. SSC has a good performance on the motion segmentation with a part of missing data. But the computation time is quite large.

Our work is most related to Local Subspace Affinity (LSA) [11], which belongs to the subspace estimation method. The overall procedure of LSA is that after the data projection of the feature points that lie in a global low-dimensional subspace. The estimation of the local subspaces can be obtained by computing its nearest neighbours (NNs) and SVD. After the local subspaces have been achieved, LSA builds the affinity matrix by using the principle angles between each local subspace. In the end the subspace segmentation is accomplished by applying the spectral clustering on the affinity matrix. The rank estimation for the global and local subspace is achieved by a model selection (MS) method. The drawback of LSA is that the number of nearest neighbours to estimate the local subspace may lead to the overestimation problem. It means that the nearest neighbours may not belong to the same subspace. This situation is more likely to happen particularly with the non-rigid or degenerated motions. The second limitation is that the rank estimation by model selection is based on the parameter  $k$  which has to be set depending on the noise and the number of motions. Thus the model selection needs to know the amount of noise as the prior knowledge as well.

### 3 Subspace Models for Motion Segmentation

In this section, we first introduce the basic idea about using the subspace models for motion segmentation. Subsequently, under the affine camera model we analyse the subspace methods for motion segmentation and show that under the subspace models it is equivalent to clustering multiple low-dimensional linear subspaces in a high-dimensional space.

### 3.1 Subspace Clustering

In practice, in order to clustering the high-dimensional data from the real-world video sequences, one needs to first look for a low-dimensional representation of the high-dimensional data. After projection of the high-dimensional data, the obtained low-dimensional subspace is embedded in a union of multiple subspaces. If we consider the union of low-dimensional projection space as a global subspace, the underlying multiple subspaces can be regarded as the local subspaces. One of the main tasks of subspace clustering is to find out the number of different local subspaces, the other is the separation of the multiple local subspaces which means that one needs to cluster the data according to different subspace, the data from the same subspace should be classified together. It means that the task of cluster the feature data according to different motions changes to cluster the data into different subspaces.

### 3.2 Multi-body Motion Segmentation with Affine Camera Model

Recently, most popular algorithms for performing the motion segmentation are assuming an affine camera model, which is useful to weak and paraperspective camera models. In this paper, the affine camera model is also used for the moving object in the video sequence. Affine camera model can transform the tracked feature points on the moving object from 3-D coordinates to 2-D position, which is formulated as [12]

$$x_{fp} = A_f \begin{bmatrix} X_F \\ 1 \end{bmatrix}, \quad (1)$$

where the  $X_F = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$ , represent the world coordinate,  $A_f = [R_{2f \times 3} | T_{2f \times 1}]$  is the  $2 \times 4$  affine transformation matrix in the  $f$  frame,  $\{x_{fp} \in R^2\}_{p=1, \dots, P}^{f=1, \dots, F}$  denote the 2-D location of tracked feature trajectory at frame  $f$ .

A general input for the motion segmentation under the affine camera model can be formulated as a data matrix containing all of the 2-D positions of tracked features, so-called the trajectory matrix

$$\begin{bmatrix} x_{11} & \dots & x_{1P} \\ \vdots & & \vdots \\ x_{F1} & \dots & x_{FP} \end{bmatrix}_{2F \times P} = \begin{bmatrix} A_1 \\ \vdots \\ A_F \end{bmatrix}_{2F \times 4} \begin{bmatrix} X_1 & \dots & X_P \\ 1 & \dots & 1 \end{bmatrix}_{4 \times N}, \quad (2)$$

One can rewrite this as

$$\begin{bmatrix} x_{11} & \dots & x_{1P} \\ \vdots & & \vdots \\ x_{F1} & \dots & x_{FP} \end{bmatrix} = W_{2F \times P}, \quad (3)$$

$$W_{2F \times P} = M_{2F \times 4} S_{N \times 4}^T, \quad (4)$$

where we call the  $M$  as the motion matrix, whereas the  $S$  is the structure matrix. The rank of the  $W$  trajectory matrix is no more than 4. As a result, the rank of the general trajectory matrix for rigid motion is at most 4.

## 4 Proposed Framework

Our proposed framework extends the framework of LSA [11] in both the global and the local parts, as shown in Fig. 2. Transformation of the trajectory matrix with the Sparse Principal Component Analysis (SPCA) is used for the global subspace estimation. For the local subspace estimation, instead of the fixed number of local nearest neighbours policy, we adopt a sparse manifold optimization from [13] to automatically extract each local low-dimensional subspaces. In Sect. 4.1, the SPCA is presented. The sparse manifold optimization technique for local subspace estimation is presented in Sect. 4.2.

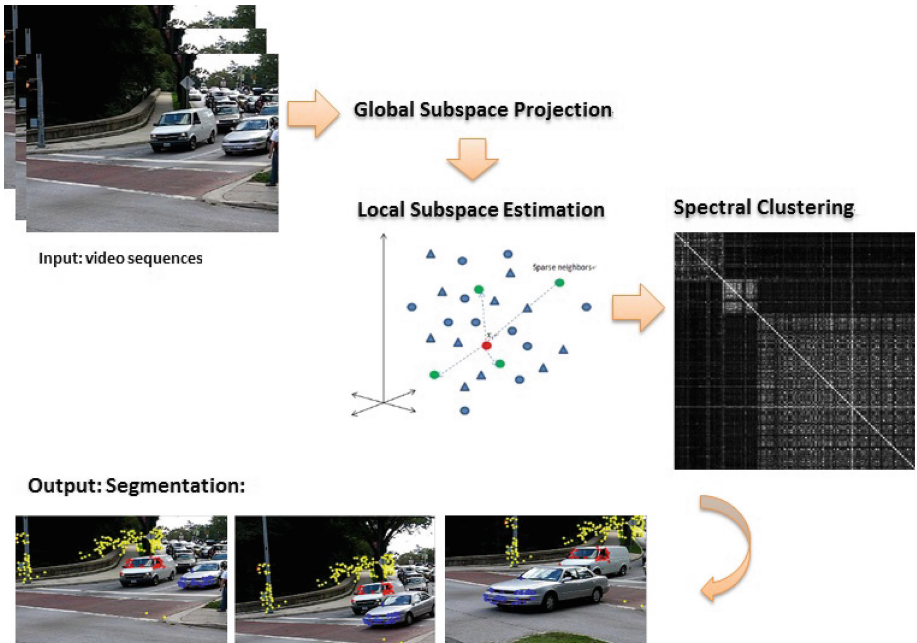


Fig. 2. Proposed framework overview.

### 4.1 Global Subspace Transformation

As described in Sect. 3.2, the dimensional of a general input trajectory data matrix for subspaces methods for motion segmentation is  $2F$ , where  $F$  denotes the number of the frames. However, the maximum rank of the subspace for one rigid motion is 4. Therefore most of the rank of the trajectory data is redundant.

This has motivated a lot of algorithms to discover a low-dimensional projection for the high-dimensional trajectory data matrix.

Assume that the trajectory matrix  $W_{2F}^P \in R^{2F}$  as Eq. 2 is given as input, from Sect. 3.2 the rank of the general motion segmentation for one rigid motion is bounded by 4. This constraints has enforced a customary dimensionality reduction procedure for trajectory matrix  $W_{2F}^P$ . Most of the other subspace methods choose to reduce the dimension of  $W_{2F}^P$  to  $m = 4n$ , where  $n$  is the number of motions. Following classical PCA we can use the singular value decomposition(SVD) to the matrix  $W$ , which decompose the  $W$  as follows:  $U_{2F \times 2F} D_{2F \times P} V_{P \times P}^T = W_{2F}^P$ . A global data transformation is then obtained by considering only the first  $m$  columns of  $V$ . In the case of SPCA we need to have a sparse construction in the matrix  $V$ , for illustration we can choose a  $l_0$ -penalty for block sparse PCA through a generalized power method proposed by [14]. After solving a sparse optimization problem on the high-dimensional motion data, we can obtain a new representation matrix denoted by  $Z^*$  of  $W_{2F}^P$  on the Stiefel manifold  $S_m^P$ . Each column of  $Z^*$  is a sparse vector  $z_i^*, i = 1, \dots, m$  represents the transformed data.

In order to enforce the sparse entries on the principal components, [15] firstly proposed the direct formulation with *Lasso* to produce sparse principal components. Given the data matrix  $W_{2F}^P$  and  $\Sigma = W^T W$  is the covariance matrix of  $W$ , the classical PCA can be formed as follows,

$$z^* = \max_{z^T z \leq 1} z^T \Sigma z, \tag{5}$$

The solutions  $z^*$  are the principal components of the data matrix  $W$ . In [15], they consider a direct reformulation to penalize the nonzero entries of the solutions  $z$ ,

$$z^* = \max_{z \in B^n} z^T \Sigma z - \gamma \|z\|_0, \tag{6}$$

with the sparsity-controlling parameter  $\gamma > 0$ , when  $\gamma = 0$ , the Eq. 6 is relative to the classical PCA problem.  $B^n$  refers to a unit Euclidean ball in  $R^n$ . Whereas, the author of [14] consider a fixed data  $k \in B^n$  which ensure that  $(x_i^T k)^2 - \gamma > 0$ , where the vector  $\{x_i \in W_{2F}^P, i = 1, \dots, P\}$ , and the Eq. 6 is changed to,

$$z^* = \max_{k \in B^p} \max_{z \in B^n} (k^T W z)^2 - \gamma \|z\|_0, \tag{7}$$

In order to obtain a accurate projected  $m$ -dimensional subspace with orthogonal vectors, we can use the block sparse PCA on a Stiefel manifold  $S_m^P$ , because the sparse principal components  $z^*$  of SPCA are not forced to be orthogonal and cannot be used to the following local subspace separation. To enforce the orthogonal principal components, the author of [14] choose the block form for PCA and solve a block SPCA based on the  $l_0$ -Penalty. Following the Eq. 6, when the  $\gamma = 0$ , we come to the classical PCA situation,

$$z^* = \max_{z \in B^n} z^T \Sigma z, \tag{8}$$

Then the author of [14] extended the Eq. 8 to the block form with a *trace function* as the following reformulation,

$$Z^* = \max_{K \in S_m^P} \max_{Z \in S_n^n} \text{Trace}(\text{diag}(K^T W Z N)^2) - \sum_{j=1}^m \gamma_j \|z_j\|_0, \quad (9)$$

where the  $m$  related to the needed transformed dimensional,  $m \leq \text{rank}(W)$ ,  $m$ -dimensional vector  $\gamma = [\gamma_1, \dots, \gamma_m]^T$  is positive. The solutions  $Z^*$  of Eq. 9 which span the dominant  $m$ -dimensional invariant subspace of the matrix  $W$  on a size fixed Stiefel manifold  $S_m^P$ . The matrix  $N = \text{Diag}(\mu_1, \mu_2, \dots, \mu_m)$  with distinct positive diagonal elements enforces the Eq. 9 to have isolated maximizers. It has been proved in the work of [14] that distinct elements on the diagonal of  $N$  enforce the loading vectors of the principal components of sparse PCA that are more orthogonal. Subsequently, Eq. 9 is completely decoupled in the columns of  $Z^*$  as follows,

$$Z^* = \max_{K \in S_m^P} \sum_{j=1}^m \max_{z_j \in S^n} (\mu_j k_j^T W z_j)^2 - \gamma_j \|z_j\|_0, \quad (10)$$

Equation 10 can be reformulated as a convex object function on the Stiefel manifold  $S_m^P$ ,

$$Z^* = \max_{K \in S_m^P} \sum_{j=1}^m \sum_{i=1}^n [\mu_j \mathbf{x}_i^T k_j]^2 - \gamma_j]_+ \quad (11)$$

where the parameters should be given under the condition,

$$(\mu_j a_i^T x_j^*)^2 > \gamma_j \quad (12)$$

As we want a global projection of trajectory matrix  $W_{2F}^P$  that is preserved with the sparse loading entries as the matrix  $Z^* \in S_m^P$ , we use the block SPCA via  $l_0$ -Penalty to obtain a sparse low-dimensional representation. In the work of [14] they have already perform an efficient solution to solve the convex objective function in Eq. 11. In the end we can achieve a  $m$ -dimensional sparse matrix  $Z^* \in S_m^P$  as the projected  $m$ -dimensional global subspace. It is equivalent to perform the segmentation of the multiple embedded affine low-dimensional subspaces (local subspaces) on the new global manifold.

### 4.2 Local Subspace Estimation

A sparse optimization technique *sparse manifold clustering and embedding* (SMCE) [13] can simultaneously estimate the neighbours of each data point from the same manifold and clustering the multiple embedded manifolds. In this work, we adopt the essential idea from SMCE for estimating the neighbours of the local subspace generated by each trajectory from the same low-dimensional local subspace.

SMCE assume that given a data point  $x_i \in R^D$  draw from a manifold  $M_l$  with dimension  $d_l$ , there exists the relative set of points  $N_i = \{x_j\}_{j \neq i}$  in  $M_l$  that contains only a few number of non-zero elements comes from the same affine subspace that passes near  $x_i$ . We call the neighbor set  $N_i$  *sparse neighbours*. This assumption

can be defined as Eq. 13, which illustrates the optimization of sparse neighbours: among all of the points  $\{x_j\}_{j \neq i}$ , the ones that are neighbours of  $x_i$  in the same manifold span a  $d_l$ -dimensional affine subspace that passes through  $x_i$ ,

$$\| [x_1 - x_i, \dots, x_N - x_i] \|_2 \leq \epsilon \quad \text{and} \quad 1^T c_i = 1 \tag{13}$$

This assumption can be solved by a simple weighted sparse optimization program under the affine constraint,

$$\begin{aligned} & \min \| Q_i c_i \|_1 \\ & \text{s.t. } \| X_i c_i \|_2 \leq \sigma, 1^T c_i = 1 \end{aligned} \tag{14}$$

where the  $Q_i$  has the diagonal elements  $\frac{\|x_{fj} - x_{fi}\|_2}{\sum_{t \neq i} \|x_{ft} - x_{fi}\|_2} \in (0, 1]$  and we can call it a proximity inducing matrix that encourage finding the close neighbours. And  $X_i$  denote the normalized new vectors, which is

$$X_i = \left[ \frac{x_{f1} - x_{fi}}{\|x_{f1} - x_{fi}\|_2} \dots \frac{x_{fP} - x_{fi}}{\|x_{fP} - x_{fi}\|_2} \right] \in R^{2F \times P-1} \tag{15}$$

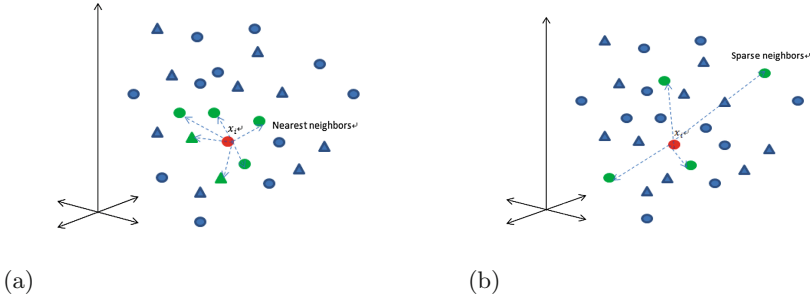
The results  $c_i^T = [c_{i1}, \dots, c_{iP}]$  have only a few non-zero entries which ideally indicate the sparse neighbours of data point  $x_i$  from the same subspace.

In our proposed framework, after a global transformation using SPCA and normalizing the projected data, we obtain a sparse global subspace  $\widehat{W}_m^P$  on the Stiefel manifold. Following the assumption of SMCE, most points and their sparse neighbours should lie on the same underlying low-dimensional subspace. We can adopt this assumption for searching the sparse representation of the closest neighbourhood of each data point in a global projected subspace. Opposite to the previous works such as LSA [11], who uses the fixed k-nearest neighbours technique, we choose to automatically estimate the sparse neighbours in an adequately large space.

A proper choice of the size for fixed k-nearest neighbours is sometimes critical and has important influence for the results of segmentation. Especially, when there are intersections between different local subspaces, the nearest neighbours can also belong to two different subspaces which could lead to the misclassification. As shown in Fig. 3, if we set the sample  $x_i$  as the observed point, Fig. 3(a) is the result of k-nearest neighbour searching. There are two triangles which should belong to the other subspace are found as the nearest neighbours of the circles, which will lead to a misclassification in the final clustering. Whereas Fig. 3(b) illustrates the sparse neighbour searching. It is clearly that if we search the nearby neighbours of each data point in the global subspace instead of looking for only the k-nearest neighbours, the estimation of the local subspace will be more accurate with avoiding the intersection of two different underlying subspace.

In order to conform the subspace geometric property constraint of each point, we choose the distance measure between two different subspaces with principal





**Fig. 3.** The selection of neighbours of k-nearest neighbours and sparse neighbours in high dimensional space. The circles and triangles represent two different subspace local samples. (a) k-nearest neighbours: the green color denote the searched nearest neighbours of observed data  $x_i$ ; (b) sparse neighbours: the green color denote the searched sparse neighbours of observed data  $x_i$  (Color figure online).

---

**Algorithm 1.** Local Subspace Estimation

---

**Require:** Projected Normalized Data Matrix  $\widetilde{W}_{m \times P} = [x'_1, x'_2, \dots, x'_P]_{m \times P}$

**Ensure:** Set of estimated local subspaces  $S_1, S_2, \dots, S_P$

- 1: **for all**  $i, j = 1, \dots, P$  **do**
  - 2:      $Pr(\theta)_j = \text{acos}(x_i^T x'_j)$
  - 3: **end for**
  - 4:  $Pr(\Theta) = [Pr(\theta)_1, Pr(\theta)_2, \dots, Pr(\theta)_P]_{P \times P}$
  - Compute inducing proximity matrix  $Q_i = \text{diag}(\frac{\theta_i}{\sum_{t \neq i} \theta_t}) \in R^{P-1 \times P-1}, i = 1, 2, \dots, P$
  - 5: **for**  $i = 1, \dots, P$  **do**
  - 6:     solve sparse optimization with parameter  $\lambda_1, \lambda_2$   
        $C_i^T = [c_{i1}, \dots, c_{iP}]^T \leftarrow \min \frac{1}{2} \|Pr(\theta)_i c_i\|_2^2 + \lambda_1 \|Q_i c_i\|_1 - \lambda_2 \mathbf{1}^T c_i$
  - 7: **end for**
  - 8: **for**  $i, j = 1, 2, \dots, p$  **do**
  - 9:      $c_i^T = [c_{i1}, \dots, c_{iP}]^T$
  - 10:     $l_i \leftarrow \text{nonzero}(c_i^T)$
  - 11: **end for**
  - 12: sparse data matrix  $\widetilde{C} = [C_1^T, C_2^T, \dots, C_P^T]_{P \times P}$   
      estimated sparse neighbours set  $L = \{l_i, i = 1, \dots, P\}$
  - 13: **for**  $i = 1, 2, \dots, P$  **do**
  - 14:     local subspace  $S_i = \widetilde{W}_{m \times P}(:, C_i^T(1 : l_i, i))$
  - 15: **end for**
  - 16: Estimated local subspaces set  $S_1, S_2, \dots, S_P$
-

angles. The principal angles between two subspaces  $S_i$  and  $S_j$  can be defined recursively with  $0 \leq \theta_1 \leq \dots \leq \theta_l \leq (\pi/2)$ ,

$$\begin{aligned}
 l &= \min(\dim(S_i), \dim(S_j)) \\
 \cos(\theta_k) &= \max_{u \in S_j, v \in S_i} u^T v = u_k^T v_k, k = 1, \dots, l \\
 \text{s.t. } \|u\| = \|v\| &= 1, u^T u_q = 0, v^T v_q = 0, q = 1, \dots, k - 1
 \end{aligned} \tag{16}$$

we can use the distance measure of two subspaces  $S_i$  and  $S_j$  with the affinity of principal angles of  $S_i$  and  $S_j$ ,

$$A(i, j) = e^{-\sum_{q=1, \dots, l} \sin^2(\theta_q)} \tag{17}$$

Thus we modify the SMCE with principal angles  $0 \leq \theta_1 \leq \dots \leq \theta_l \leq (\pi/2)$ ,

$$X_i = \theta_i \tag{18}$$

where the  $\cos(\theta_i) = \max_{u \in S_j, v \in S_i} u^T v = u_k^T v_k, k = 1, \dots, l$ ,  $S_j$  is the subspace generated by the data point  $x_j$  and  $S_i$  is generated by  $x_i, j \neq i$ .

Subsequently we can modify the optimization program in Eq. 14 and define the weight proximity inducing weight matrix and solving the Eq. 14 by the argued the Lagrange multipliers [16], which is

$$\begin{aligned}
 \min & \frac{1}{2} \|\theta_i c_i\|_2^2 + \lambda \|Q_i c_i\| \\
 \text{s.t. } & 1^T c_i = 1,
 \end{aligned} \tag{19}$$

where  $Q_i = \text{diag}(\frac{\theta_i}{\sum_{t \neq i} \theta_t}) \in R^{N-1 \times N-1}, \theta_i = \text{acos}(\max_{u \in S_j, v \in S_i} u^T v) = \text{acos}(u_i^T v_i), \theta_t = \text{acos}(\max_{u \in S_i, v \in S_i} u^T v) = \text{acos}(u_i^T v_t), t \neq j$ . The procedures of the local subspace estimation are summarized in Algorithm 1.

With solving the problem in Eq. 19, the number of the sparse neighbours is obtained from the sparse solutions  $C_i$ , which indicates the neighbours of point  $x_i$  from the same subspace. The estimation of the local subspace can be achieved by simply extracting the neighbours for each data point. Subsequently, we can use the affinity measure to compute the similarity between each pair of estimated local subspaces and build the symmetric affinity matrix. There are amount of measure techniques for different subspaces like subspace euclidean distances or principal angles. In the end we can easily perform a simple spectral clustering on the built affinity matrix.

### 4.3 Complete Procedure

Let  $W_{2F \times P} = [x_1, x_2, \dots, x_P]$ , where each  $x_i = [x_{1i}, \dots, x_{Fi}]^T$  represent a tracked feature trajectory throughout  $F$  frames. All of the trajectories are  $\{x_i \in R^D\}$  drawn from  $n$  different subspaces with dimensions  $\{d_j \in R^D, j = 1, \dots, n\}$ . In general, our proposed subspace-based motion segmentation method can be written

with the 3 main steps for clustering the feature data from a union of multiple affine subspaces: (1) Transform the input trajectories into a sparse global subspace with Sparse PCA; (2) Estimate the local subspace with the number of sparse neighbours and the principal angles; (3) Build the similarity matrix with principal angles and clustering the matrix with k-means. The final output is the labelled moving objects. The overall motion segmentation algorithm is summarized in Algorithm 2.

---

**Algorithm 2.** Motion segmentation using sparse optimization

---

**Require:** Data Matrix  $W_{2F \times P} = [x_1, x_2, \dots, x_P]_{2F \times P}$ , number of motions  $n$

**Ensure:** A Set of Labels  $[1, 2, \dots, n]$

---

**Step 1. Global Projection:**  $m = 4n, W'_{m \times P} \leftarrow W_{2F \times P}$

Sparsity-controlling parameter  $\gamma = [\gamma_1, \gamma_2, \dots, \gamma_m]^T \geq 0, N = [\mu_1, \dots, \nu_m] \geq 0$

1: **for all**  $i = 1, \dots, P$  **do**

2:  $x'_i \leftarrow SPCA(x_i, \gamma, N)$

3: **end for**

4:  $W'_{m \times P} = [x'_1, x'_2, \dots, x'_P]_{m \times P}$

5: Normalization  $\widetilde{W}_{m \times P} = \text{normalize}(W'_{m \times P})$

**Step 2. Local Subspace Estimation:**

6: **for all**  $i, j = 1, \dots, P$  **do**

7: run Alg.1 to estimate the local subspaces  $\mathbf{S} = \{S_1, S_2, \dots, S_P\}$  for each projected trajectory  $x'_i$

8: **end for**

**Step 3. Perform Spectral Clustering**

9: **for**  $i, j = 1, 2, \dots, P$  **do**

10:  $A(i, j) = Pr(B_i, B_j)$

11: **end for**

12: Perform spectral clustering on  $A$  to obtain a set of  $n$  Labels

---

## 5 Experimental Results

In this section we evaluate our method on a standard real-world video sequence benchmark, the Hopkins 155 dataset [17]. We have compared with other state-of-art motion segmentation approaches. We have assumed for all the methods that the number of the motions has already given.

We test all the algorithms on the full original Hopkins 155 dataset [17] with no missing trajectories. The database from the Hopkins 155 composed of 120 sequences with 2 motions and 35 sequences with 3 motions. There are 3 different kinds of motions in the Hopkins 155 dataset: traffic, checkerboard and articulated. As a pre-processing step, the feature points that tracked through all over the sequences have already been obtained. Moreover, all of the errors in tracking were already corrected for each sequence. Hence, this experiment exists



**Fig. 4.** Comparison of Our approach with ground truth (*the last row*) and the other approaches *the first row*: LSA [11]; *Second row*: MSMC [6], *Third row*: Our Method on the frames 1, 8, 15 and 20 of the *cars02-07* sequence from the Hopkins 155 dataset [17].

no missing entries in the feature trajectories. Most people use the Hopkins 155 dataset for testing the performance of accuracy and computation times.

We have tested SSC [10], LSA [11], RANSAC [7], GPCA [8], MSMC [6], LLMC [18] and our method on the checkboard, traffic and articulated sequences, because of MSMC based on the affinity we have not compare it to the others in the checkboard sequences. The parameter  $k$  in LSA has been set to  $10^{-6}$  and the number of nearest neighbours is fixed with 6. The threshold for subspace fitting needed for RANSAC method has chosen to be 0.00002. The sparsity controls parameter  $\lambda$  for SMCE [13] in our method is set to be 20. For all of the methods, the data has been projected into the  $4n$  global subspace according to the number of motions. We have computed the average and median misclassification error for comparison, as shown in Tables 1, 2, and 3. These numbers show that our segmentation results are comparable with other state-of-art motion segmentation results, and in many cases exceed them in terms of segmentation accuracy. Figure 4 shows the

**Table 1.** Mean and median of the missclassification (%) on the Hopkins155 database with 2 motions.

Method	GPCA	ALC	SSC	MSMC	LLMC	LSA	Our method
<b>Articulated 11 sequences</b>							
Mean	2.88	10.70	0.62	2.38	5.23	4.10	0.55
Median	0.00	0.95	0.00	0.00	1.30	0.00	0.00
<b>Traffic: 31 sequences</b>							
Mean	1.41	1.59	0.02	0.06	3.65	5.43	0.59
Median	0.00	1.17	0.00	0.00	0.33	1.48	0.00
<b>Checkerboard 78 sequences</b>							
Mean	6.09	1.55	1.12	NaN	4.65	2.57	1.42
Median	1.03	0.29	0.00	NaN	0.11	0.27	0.27
<i>All 120 sequences</i>							
Mean	4.59	2.40	0.82	NaN	4.44	3.45	1.11
Median	0.38	0.43	0.00	NaN	0.24	0.59	0.00

**Table 2.** Mean and median of the missclassification (%) on the Hopkins155 database with 3 motions.

Method	GPCA	ALC	SSC	MSMC	LLMC	LSA	Our method
<b>Articulated 2 sequences</b>							
Mean	16.85	21.08	1.91	1.42	9.38	7.25	5.32
Median	16.85	21.08	1.91	1.42	9.38	7.25	5.32
<b>Traffic: 7 sequences</b>							
Mean	19.83	7.75	0.58	0.16	7.79	25.07	4.74
Median	19.55	0.49	0.00	0.00	5.47	23.79	4.04
<b>Checkerboard 26 sequences</b>							
Mean	31.95	5.20	2.97	NaN	12.01	5.80	3.05
Median	32.93	0.67	0.27	NaN	9.22	1.77	0.77
<i>All 35 sequences</i>							
Mean	28.66	6.69	2.45	NaN	11.02	9.73	3.49
Median	28.66	0.67	0.20	NaN	6.81	2.33	1.11

qualitative segmentation examples from the Hopkins 155 dataset. We can infer that our method segments the foreground moving object successfully in comparing with the ground truth and other algorithms.

We also present the run-time of our method, SSC [10], and ALC [9] in Table 4. Comparing with the SSC [10], the performance of our method is better than SSC

**Table 3.** Mean and median of the missclassification (%) on the whole Hopkins155 database with both 2 and 3 motions.

Method	GPCA	ALC	SSC	LLMC	LSA	Our method
<i>All 155 sequences</i>						
Mean	10.34	3.56	1.24	5.93	4.94	1.65
Median	2.54	0.50	0.00	0.63	0.90	0.32

**Table 4.** Run-Time (%) for the whole Hopkins155 database with both 2 and 3 motions.

Method	ALC	SSC	Our method
Run-time [s]	88831	14500	<b>14021</b>

particularly on the articulated sequence of 2 motions. With only a little loss of accuracy on the other sequence, we save the computation time w.r.t. the SSC [10].

## 6 Conclusions

We have proposed a feature-based framework for the problem of segmenting different types of the moving objects from a video sequence with combining two sparse subspace optimization methods SPCA [19] and SMCE [13]. The SPCA performs a data projection from a high-dimensional subspace to a low-dimensional global manifold with sparse entries, which ensures the interpretability and accuracy. Simultaneously, we adopt the idea of SMCE that search the sparse closest neighborhood set for each local embedded subspace generated by each trajectory, which efficiently solve the intersection or overestimation problem in LSA framework [11]. The experiments demonstrate that the low misclassification error of our approach on the Hopkins 155 dataset [17], outperforming most of the popular approaches. The limitation of our work is that the number of motions is considered as a prior knowledge. In the future work, our goal is to perform the estimation of the number of motions in the framework as well. Furthermore, we will derive a robust optimization method that can deal with the corrupted trajectory and missing data.

**Acknowledgments.** The work is funded by the ERC-Starting Grant (‘DYNAMIC MINVIP’). The authors gratefully acknowledged the support.

## References

1. Lee, Y.J., Kim, J., Grauman, K.: Key-segments for video object segmentation. In: 2011 IEEE International Conference on Computer Vision (ICCV), IEEE, pp. 1995–2002 (2011)
2. Shi, J., Malik, J.: Motion segmentation and tracking using normalized cuts. In: 1998 Sixth International Conference on Computer Vision, IEEE, pp. 1154–1160 (1998)

3. Tomasi, C., Kanade, T.: Detection and Tracking of Point Features. School of Computer Science, Carnegie Mellon University, Pittsburgh (1991)
4. Zhou, H., Yuan, Y., Shi, C.: Object tracking using sift features and mean shift. *Comput. Vis. Image Underst.* **113**, 345–352 (2009)
5. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). *Comput. Vis. Image Underst.* **110**, 346–359 (2008)
6. Dragon, R., Rosenhahn, B., Ostermann, J.: Multi-scale clustering of frame-to-frame correspondences for motion segmentation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) *ECCV 2012, Part II*. LNCS, vol. 7573, pp. 445–458. Springer, Heidelberg (2012)
7. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**, 381–395 (1981)
8. Vidal, R., Ma, Y., Sastry, S.: Generalized principal component analysis (GPCA). *IEEE Trans. Pattern Anal. Mach. Intell.* **27**, 1945–1959 (2005)
9. Ma, Y., Derksen, H., Hong, W., Wright, J.: Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**, 1546–1562 (2007)
10. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, IEEE, pp. 2790–2797 (2009)
11. Yan, J., Pollefeys, M.: A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3954, pp. 94–106. Springer, Heidelberg (2006)
12. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, New York (2003)
13. Elhamifar, E., Vidal, R.: Sparse manifold clustering and embedding. In: NIPS, pp. 55–63 (2011)
14. Journée, M., Nesterov, Y., Richtárik, P., Sepulchre, R.: Generalized power method for sparse principal component analysis. *J. Mach. Learn. Res.* **11**, 517–553 (2010)
15. d’Aspremont, A., El Ghaoui, L., Jordan, M.I., Lanckriet, G.R.: A direct formulation for sparse pca using semidefinite programming. *NIPS* **16**, 41–48 (2004)
16. Tibshirani, R.: Regression shrinkage and selection via the lasso. *J. Roy. Stat. Soc. Ser. B (Methodological)* **58**, 267–288 (1996)
17. Tron, R., Vidal, R.: A benchmark for the comparison of 3-d motion segmentation algorithms. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, IEEE, pp. 1–8 (2007)
18. Goh, A., Vidal, R.: Segmenting motions of different types by unsupervised manifold clustering. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, IEEE, pp. 1–6 (2007)
19. Zou, H., Hastie, T., Tibshirani, R.: Sparse principal component analysis. *J. Comput. Graph. Stat.* **15**, 265–286 (2006)