Motion Segmentation Using Global and Local Sparse Subspace Optimization

Michael Ying Yang, Hanno Ackermann, Weiyao Lin, Sitong Feng, and Bodo Rosenhahn

Abstract

In this paper, we propose a new framework for segmenting feature-based moving objects under the affine subspace model. Since the feature trajectories are high-dimensional and contain the noise, we first apply the sparse PCA to represent the original trajectories with a low-dimensional global subspace, which consists of the orthogonal sparse principal vectors. Then, the local subspace separation is obtained using automatically searching the sparse representation of the nearest neighbors for each projected data. In order to refine the local subspace estimation and deal with the missing data problem, we propose an error estimation function to encourage the projected data that span a same local subspace to be clustered together. Finally, the segmentation of different motions is achieved through the spectral clustering on an affinity matrix, which is constructed with both the error estimation and the sparse neighbor optimization. We evaluate our proposed framework by comparing it to other motion segmentation algorithms. Our method achieves improved performance on state-of-the-art benchmark datasets.

Introduction

Motion segmentation is an essential task for understanding the dynamic scenes and other computer vision applications [1].[2]. Particularly, motion segmentation aims to decompose a video into different regions according to different moving objects that tracked throughout the video. In case of feature extraction for all the moving objects from the video, segmentation of different motions is equivalent to segment the extracted feature trajectories into different clusters. One example of feature-based motion segmentation is presented in Figure 1.

Generally, the algorithms of motion segmentation are classified into two categories [4]: affinity-based methods and subspace-based methods. The affinity-based methods focus on computing the correspondences of each trajectory pair, whereas the subspace-based approaches use multiple subspaces to model the multiple moving objects in the video, and the segmentation of different motions is accomplished through subspace clustering. Recently, some affinity-based methods [4] [5] are proposed to cluster the trajectories with unlimited number of missing data. However, the computational cost is very high. Whereas, the subspace-based methods [6] [7] have been developed to reconstruct the missing trajectories with their sparse representation. The drawback is that they are sensitive to the real video which contains a large number of missing trajectories. Most of the existing subspace-based methods still fall their robustness for handling missing features. Thus, there is an intense demand to explore a new subspace-base algorithm that can not only segment multiple kinds of motions, but also handle the missing and corrupted trajectories from the real video.

Contributions

We propose a new framework with subspace models for segmenting different types of moving objects from a video under the affine camera. We cast the motion segmentation as a two stage subspace estimation: the global and local subspace estimation. Sparse PCA [8] is adopted for optimizing the global subspace in order to defend the noise and outliers. Meanwhile, we seek a sparse representation for the nearest neighbors in the global subspace for each data point that span a same local subspace. In order to solve the missing data problem and refine the local subspace estimation, we build the affinity graph for the spectral clustering with a novel error estimation function. To the best of our knowledge, our framework is the first one to simultaneously optimize the global and local subspace with sparse representation.

The remaining sections are organized as follows. The related works are discussed in the next Section, followed by the basic subspace models for motion segmentation. The proposed approach is described in detail followed by the experimental results are presented leading to the is conclusions.

Figure 1. Example results of the motion segmentation on the real traffic video *cars9.avi* from the Hopkins 155 dataset [3].

Michael Ying Yang is with the University of Twente, ITC, Hengelosestaat 99, Enschede, The Netherlands (michael.yang@utwente.nl).

Hanno Ackermann, Sitong Feng, and Bodo Rosenhahn are with Leibniz University Hannover.

Weiyao Lin is with the Shanghai Jiao Tong University (Corresponding Author).

Photogrammetric Engineering & Remote Sensing
Vol. 83, No. 11, November 2017, pp. 769–778.
0099-1112/17/769–778

doi: 10.14358/PERS.83.10.769

Related Work

During the last decades, either the subspace-based techniques [6] [7] or the affinity-based methods [4] [5] have been receiving an increasing interest on segmentation of different types of motions from a real video.

Affinity-based methods [5] use the distances of each pair of feature trajectories as the measurement to build the affinity matrix based on a translational motion model. This method can segment motions with unlimited number of missing or incomplete trajectories, which means they are robust to the video with occlusions or moving camera problems. Another approach which is based on the affinity is called Multi-Scale Clustering for Motion Segmentation (MSMC) [4]. Based on the split and merge, MSMC uses the correspondences of two features between two frames to segment the different motions with many missing data. One of the general problems of affinity-based method is highly time-consuming.

In Subspace-Based Methods the existing work based on subspace models can be divided into four categories: algebraic, iterative, sparse representation, and subspace estimation.

Algebraic approaches, such as Generalized Principal Component Analysis (GPCA) [9], use the polynomials fitting and differentiation to obtain the clusters. GPCA can segment the rigid and non-rigid motions effectively. However, when the number of moving objects in the video increases, its computational cost increases and the precision decreases at the same time. The general procedure of an iterative method contains two main aspects: finding the initial solution and refining the clustering results to fit each subspace model. RANdom SAmple Consensus (RANSAC) [10] selects randomly the number of points from the original dataset to fit the model. RANSAC is robust to the outliers and noise, but it requires good initial parameter selection. Specifically, it computes the residual of each point to the model within a threshold. Sparse Subspace Clustering (SSC) [6] is one of the most popular motion segmentation methods based on the sparse representation. SSC exploits the fact that each point can be linearly represented with a sparse combination of the rest of other data points. The limitation is that the computational cost of SSC is very high. Another popular algorithm based on the sparse representation is Agglomerate Lossy Compression (ALC) [7], which uses compressive sensing on the subspace model to segment the video with the missing trajectories. However, ALC cannot guarantee to find the global maximum with the greedy algorithm.

Our work combines the subspace estimation and sparse representation methods. The subspace estimation algorithms, such as Local Subspace Affinity (LSA) [11LSA], first projects the original data set with a global subspace. Then, the projected global subspace is separated into multiple local subspaces with K-nearest neighbors (KNN). After calculating the affinities of different estimated local subspaces with principle angles, the final clusters are obtained through spectral clustering. The issue is that the KNN policy may overestimate the local subspaces due to noise and improper selection of the number K, which is determined by the rank of the local subspace. LSA uses the model selection (MS) [12] to estimate the rank of global and local subspaces, but the MS is sensitive to the noise level.

Multi-Body Motion Segmentation with Subspace Models

In this section, we introduce the motion structure under the affine camera model. Subsequently, we show that under the affine model segmentation of different motions is equivalent to separate multiple low-dimensional affine subspaces from a high-dimensional space.

Affine Camera Model

Most of the motion segmentation algorithms assume the affine camera model [LSA], which is the orthographic camera model and has a simple mathematical form. Under the affine camera, the general procedure for motion segmentation is started from translating the 3-D coordinates of each moving object to its 2-D locations in each frame. Assume that $\left\{x_{fp}\right\}_{f=1,\ldots,F}^{p=1,\ldots,P} \in R^2$ represents one 2D tracked feature point p of one moving object at frame f, its corresponding 3D world coordinate is $\left\{X_p\right\}_{p=1,\ldots,p} \in R^3$. The pose of the moving object at frame f can be represented with R_f, T_f, where R_f and T_f are the rotation and the translation, respectively. Therefore, each 2D point x_{fp} can be described as [11]:

$$x_{fp} = \left[R_f T_f\right] X_p = A_f X_p \tag{1}$$

where $A_f = \left[R_f T_f\right] \in R^{2\times 4}$ is the affine transformation matrix at frame f.

Subspace Models for Motion Segmentation Under the Affine View

The general input for the subspace-based motion segmentation under the affine camera can be formulated as a trajectory matrix containing the 2D positions of all the feature trajectories tracked throughout all the frames. Given 2-D locations $\left\{x_{fp}\right\}_{f=1,\ldots,F}^{p=1,\ldots P} \in R^2$ of the tracked features on a rigid moving object, the corresponding trajectory matrix can be formulated as:

$$W_{2F\times P} = \begin{bmatrix} x_{11} & \cdots & x_{1P} \\ \vdots & \vdots & \vdots \\ x_{F1} & \cdots & x_{FP} \end{bmatrix} \tag{2}$$

Under the affine model, the trajectory matrix $W_{2F\times P}$ can be further reformulated as

$$W_{2F\times P} = \begin{bmatrix} A_1 \\ \vdots \\ A_F \end{bmatrix}_{2F\times 4} \begin{bmatrix} X_1 & \cdots & X_P \\ 1 & \cdots & 1 \end{bmatrix} \tag{3}$$

We can rewrite this equation as follows

$$W_{2F\times P} = M_{2F\times 4} S^T_{P\times 4} \tag{4}$$

where $M_{2F\times 4}$ is called motion matrix, whereas $S_{P\times 4}$ is structure matrix. According to Equation 4, the rank of trajectory matrix $W_{2F\times P}$ of a rigid motion is no more than 4. The global subspace transformation is to reduce the dimensionality of the trajectory matrix with a low-dimension representation. Then, each projected trajectory from the global subspace lives in a local subspace. The task of multi-body motion segmentation is to separate these underlying local subspaces from the global subspace, which means the segmentation of different motions is related with segmenting different subspaces.

Proposed Framework

Our proposed framework extends the LSA [11] with sparse optimization for both the global and local parts. As shown in Figure 2, given the trajectory matrix, we first transform it into a global subspace with sparse PCA [8], which is robust to noise and outliers. Instead of using the KNN estimation, we use the sparse neighbors to automatically find the projected data points spanning a same subspace. To correct the overestimation, we propose an error estimation function to build the affinity matrix for spectral clustering.

Global Subspace Transformation

In order to contain the orthogonality of projected vectors in the global subspace, we apply the generalized power method for sparse PCA [13] to transform the global subspace. Given the trajectory matrix $W_{2F\times P}=\left[w_1\ldots,w_F^T\right]$, where $w_f\in R^{2\times P}, f=1,\ldots,F$ contains all the tracked P 2D feature points in each frame f. We can consider a direct single unit form as follows to extract one sparse principal component $z^*\in R^P$ [8][13].

$$z^*(\gamma)=\max_{y\in B^P}\max_{z\in B^{2F}}\left(y^TWz\right)^2-\gamma\|z\|_0 \tag{5}$$

where y denotes an initial fixed data point from the unit Euclidean sphere $B^P=\{y\in R^P\mid y^Ty\le 1\}$, and $\gamma>0$ is the sparsity controlling parameter. If project dimension is m,$1<m<2F$, there are more than one sparse principal components needed to be extracted in order to enforce the orthogonality for the projected principal vectors. [13] extends Equation 5 to block form with a trace function:

$$Z^*(\gamma)=\max_{Y\in S_m^P}\max_{Z\in\left[S^{2F}\right]^m}Tr\left(Diag\left(Y^TWZN\right)^2\right)-\sum_{j=1}^{m}\gamma_j\|z_j\|_0 \tag{6}$$

where $\gamma=[\gamma_1,\ldots,\gamma_m^T]$ is a positive m-dimensional sparsity controlling parameter vector, and parameter matrix $N=Diag(\mu_1,\mu_2,\ldots,\mu_m)$ with setting distinct positive diagonal elements enforces the vectors Z^* to be orthogonal, $S_m^P=\{Y\in R^{P\times m}\mid Y^TY=I_m\}$ represents the *Stiefel manifold*[1]. Subsequently, Equation 6 is completely decoupled in the columns of $Z^*(\gamma)$ as follows

$$Z^*(\gamma)=\max_{Y\in S_m^P}\sum_{j=1}^{m}\max_{Z_j\in S^{2F}}\left(\mu_jy_j^TWz_j\right)-\gamma_j\|z_j\|_0. \tag{7}$$

Obviously, the objective function in Equation 7 is not convex. But the solution $Z^*(\gamma)$ can be obtained after solving a convex problem:

$$Y^*(\gamma)=\max_{Y\in S_m^P}\sum_{j=1}^{m}\sum_{i=1}^{F}\left[\left(\mu_jw_j^Ty_j\right)^2-\gamma_j\right]_+ \tag{8}$$

which under the constraint that all $\gamma_j>\mu_j^2\max_i\|w_i\|_2^2$. In [13], a gradient scheme has been proposed to efficiently solve the convex problem in Equation 8. Therefore, the sparsity pattern $\mathbf{I}$ for the solution Z^* is defined by Y^* under the following criterion.

$$\mathbf{I}=\begin{cases}active; & \left(\mu_jw_i^Ty_j^*\right)^2>\gamma_j;\\ 0; & otherwise\end{cases} \tag{9}$$

The seeking sparse vectors $Z^*\in S_m^P$ are obtained after iteratively solving Equation 8. After normalization, the global projected subspace $W_{m\times P}=normalize(Z^*)^T$ is achieved, which is embedded with multiple orthogonal underlying local subspaces.

Local Subspace Estimation

In order to cluster the different subspaces according to the different moving bodies, we need to find out the multiple underlying local subspaces from the global subspace. Generally, the estimation of different local subspaces can be addressed as the extraction of different data sets, which contain only the projected trajectories from the same subspace. One traditional approach is the local sampling [11], which uses the KNN. Specifically, the underlying local subspace spanned by each projected data is found by collecting each projected data point and its corresponding K nearest neighbors, which are calculated by the distances [11][14]]. However, the local sampling cannot ensure that all the extracted K nearest neighbors span one same subspace, which means an overestimation, especially for the video that contains many degenerated motions or missing data. Moreover, [15] shows that the selection of number K is sensitive to the rank estimation. To avoid the searching for only nearest neighbors and solve the overestimation problem, we adopt the sparse nearest neighbor optimization to automatically find the set of the projected data points that span a same local subspace.

The assumption of sparse nearest neighbors is derived from SMCE [16], which can robustly cluster the data point from a same manifold. Given a random data point x_i that draws from a manifold M_l with dimension d_l, under the SMCE assumption, we can find a relative set of points $\mathcal{N}_i=x_j, j\neq i$ from M_l but contain only a small number of non-zero elements that pass through x_i. This assumption can be mathematically defined as:

$$\left|c_i\left[x_1-x_i,\ldots,x_P-x_i\right]\right\|_2\le\varepsilon,\ s.t.\mathbf{1}^Tc_i=\mathbf{1} \tag{10}$$

1. Stiefel manifold: the Stiefel manifold $V^k(R^n)$ is the set of all orthonormal k-frames in R^n.

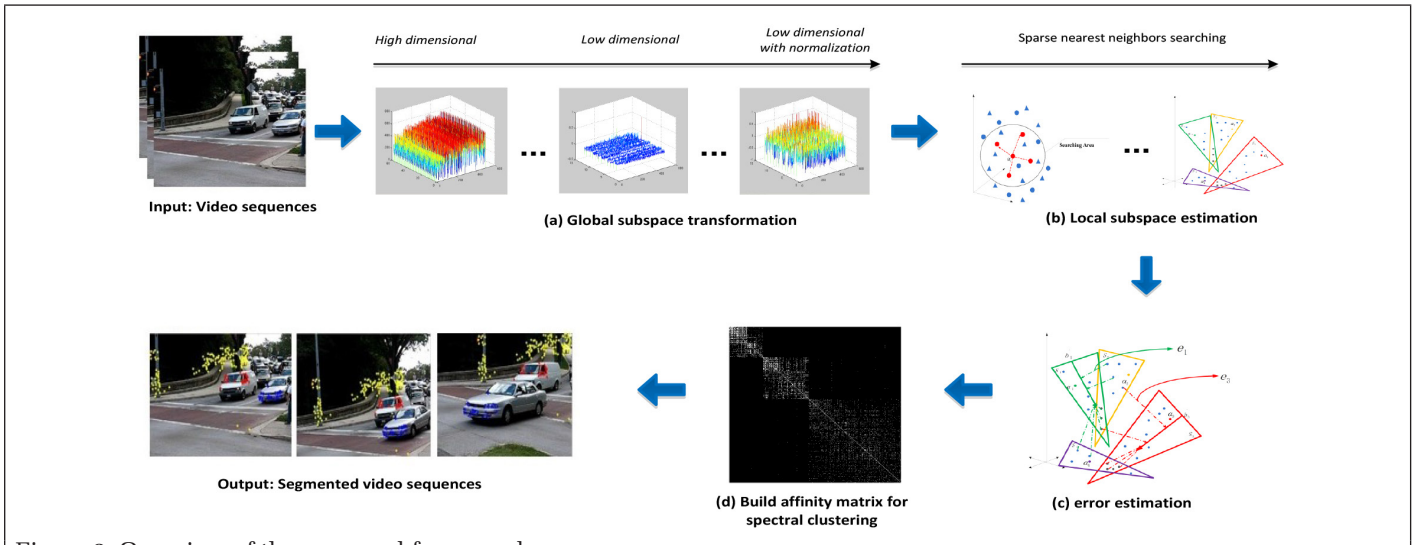


Figure 2. Overview of the proposed framework.

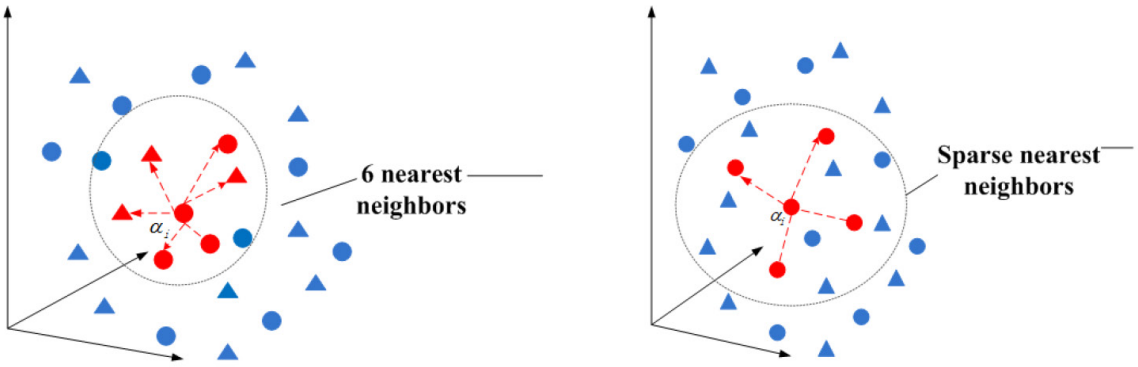


Figure 3. Illustration of six-nearest neighbor policy and sparse nearest neighbor policy. The circles and triangles represent the data points from two different local subspaces, respectively. The red points denote the estimated neighbors for the observed data α_i from the same local subspace under the determinate searching area.*i*

where c_i contains only a few non-zero entries that denote the indices of the data point that are the sparse neighbors of x_i from the same manifold, $\mathbf{1}^T c_i = \mathbf{1}$ is the affine constraint and P represents the number of all the points in the entire manifold.

We apply the sparse neighbor estimation to find the underlying local subspaces in the transformed global subspace. As illustrated in Figure 3, with the 6-nearest neighbor estimation, there are four triangles selected to span the same local subspace with observed data α_i. Whereas the sparse neighbor estimation looks for only a small number of data point close to α_i, in this way most of the intersection area between the different local subspaces can be eliminated. In particular, we constrain the searching area of the sparse neighbors for each projected trajectory from the global subspace with the normalized subspace inclusion (NSI) distances [17]. NSI can give us a robust measurement between the orthogonal projected vectors based on their geometrical consistency, which is formulated as

$$NSI_{ij} = \frac{tr\left\{\alpha_i^T \alpha_j \alpha_j^T \alpha_i\right\}}{\min\left(\dim\left(\alpha_i\right), \dim\left(\alpha_j\right)\right)} \tag{11}$$

where the input is the projected trajectory matrix $W_{m\times P} = [\alpha_1, \ldots, \alpha_P]$, and α_i, α_j, i, $j = 1, \ldots, P$ represent two different projected data. The reason of using NSI distances to constrain the sparse neighbors searching area is the geometric property of the projected global subspace. Nevertheless the data vectors which are very far away from α_i definitely cannot span the same local subspace with α_i.

Furthermore, all the NSI distances are stacked into a vector $X_i = [NSI_{i1}, \ldots, NSI_{iP}]^T$, the assumption from SMCE in Equation 10 can be solved with a weighted sparse L_1 optimization under the affine constraint, which is formulated as follows:

$$\begin{aligned} &\min \left\| Q_i c_i \right\|_1 \\ &s.t. \left\| X_i c_i \right\|_2 \le \epsilon, 1^T c_i = 1 \end{aligned} \tag{12}$$

where Q_i is a diagonal weight matrix $Q_i = \frac{\exp(X_i / \sigma)}{\exp(\Sigma X_{it}) / \sigma} \in (0,1]), \sigma > 0$
$t \neq i$
.

The effect of the positive-definite matrix Q_i is to encourage the selection of the closest points for the projected data α_i with a small weight, while the points that are far away to α_i will have a larger weight. The optimization problem is solved with Alternating Direction Method of Multipliers (ADMM) [18].

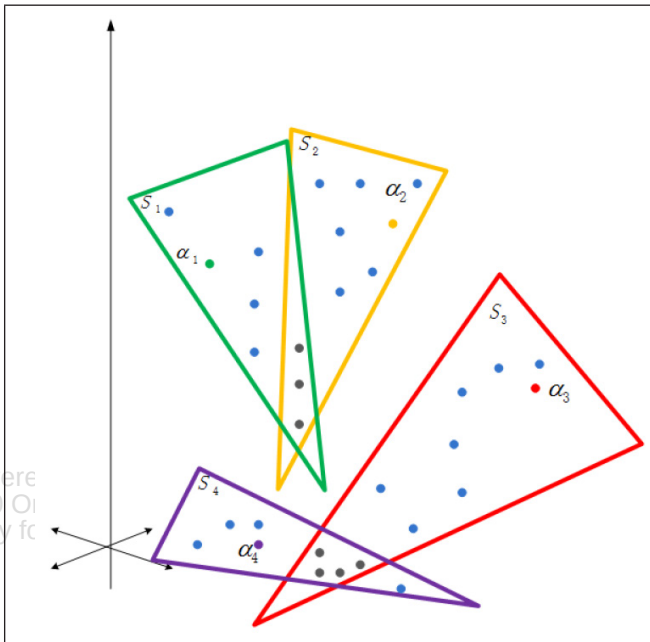


Figure 4. The geometrical illustration of incorrect local subspace estimation with sparse neighbors. S_1, S_2, S_3, S_4 are four estimated local subspaces spanned by the observed data α_1, α_2, α_3, α_4, respectively.

We obtain the sparse solution $C_{P\times P} = [c_1, \ldots, c_P]^T$ with a few number of non-zero elements that contain the connections between the projected data point and its estimated sparse neighborhood. As shown in SMCE [16], in order to build the affinity matrix with sparse solution $C_{P\times P}$, one can formulate a sparse weight matrix $\Omega_{P\times P}$ with vector ω_i, with

$\omega_i = 0, \omega_{ij} = \frac{c_{ij} / X_{ij}}{\sum_{t \neq i} c_{it} / X_{ti}}, j \neq i$. The weight matrix $\Omega_{P\times P}$ contains only a few non-zero entries in column. These non-zero entries give the indices of all the estimated sparse neighbors and the distances between them. We collect each data α_i and its estimated sparse neighbors N_i into one local subspace S_i according to the non-zero elements of *i*.

Error Estimation

The local subspace estimation after the sparse neighbor searching is illustrated in Figure 4. The estimated local subspaces are not completely spanned by each observed data

and its corresponding sparse neighborhood. There are some neighbors spanning two different local subspaces, so-called overlapping estimation problem.

In order to resolve the overlapping estimation problem, we propose the following error function:

$$e_{it} = \left\| \left(I - \beta_i \beta_i^{+} \right) \alpha_t \right\|_2^2, t = 1, \ldots, P \tag{13}$$

where $\beta_i \in R^{m \times m}$ is the basis of estimated local subspace S_i, $m_i = rank(S_i)$, β_i^{+} is the *Moore-Penrose inverse* of β_i, and $I \in R^{m \times m}$ is the identity matrix. The geometrical meaning of this error function e_{it} is the distance between the estimated local subspace and the projected data. If the projected data α_t comes from the local subspace S_i, the corresponding error e_{it} should have a very small value. After computing the corresponding error vector $e_i = [e_{i1}, \ldots, e_{iP}]$ for each estimated local subspace S_i, an error matrix $\mathbf{e}_{P \times P} = [e_1, \ldots, e_P]$ is constructed, which contains the connection between the projected data spanning a same local subspace.

By combining the estimated error matrix $\mathbf{e}_{P \times P}$ and the sparse weight matrix $\Omega_{P \times P}$, we construct the affinity graph $G=(V,E)$. The nodes V represent all the projected data points and edges E denote the distances between them. In the affinity graph, the connection between two nodes α_i and α_j is determined by both e_{ij} and ω_{ij}. Therefore, the constructed affinity graph contains only several connected elements, which are related to the data points spanning a same subspace. Formally, the adjacency matrix of the affinity graph is formulated as follows

$$A[i] = |\omega_i| + |e_i|$$
$$\mathcal{A} = \begin{bmatrix} A[1] & 0 & \cdots & 0 \\ 0 & A[2] & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A[P] \end{bmatrix} \Gamma \tag{14}$$

where the $\Gamma \in R^{p \times p}$ is an arbitrary permutation matrix. The normalized spectral clustering [19] is performed on the symmetric matrix A, and the final clusters are obtained with each cluster representing one moving object.

Experimental Results

Our proposed framework is evaluated on the Hopkins 155 dataset [3] and the Freiburg-Berkeley Motion Segmentation Dataset [5].

Implementation Details

Most popular subspace-based motion segmentation methods [6][11][7][4][5][assume that the number of motions has been known already. For the Hopkins 155 dataset, we give the exactly number of clusters according to the number of motions, while for the Berkeley dataset we set the number of clusters with seven for all the test sequences. In this paper, the area for searching the sparse neighbors is set to 20. In our experiments, we have applied the PCA and sparse PCA for evaluating the performance of our framework on estimating the multiple local subspaces from a general global subspace with dimension m=5. The sparsity controlling parameter for sparse PCA is set to γ=0.01 and the distinct parameter vector $(\mu_1, \ldots, \mu_m)$ is set to $[1/1, 1/2, \ldots, 1/m]$.

The Hopkins 155 Dataset

The Hopkins 155 dataset [3] contains three different kinds of sequences: checkerboard, traffic, and articulated. For each of them, the tracked feature trajectories are provided in the ground truth and the missing features are removed as well, which means the trajectories in the Hopkins 155 dataset are fully observed and there is no missing data. We have computed the average and median misclassification error for our method and other state-of-the-art methods: SSC [6], LSA [11], ALC [7], and MSMC [4], as shown in Table 1, Table 2, and Table 3. Table 4 shows the computational cost of our method comparing with two sparse optimization based methods: ALC and SSC.

Table 1. Mean and median of the misclassification (%) on the Hopkins 155 dataset with two motions.

Method	ALC	SSC	MSMC	LSA	Our$_{pca}$	Our$_{spca}$
Articulated, 11 sequences						
mean	10.70	0.62	2.38	4.10	2.67	0.55
median	0.95	0.00	0.00	0.00	0.00	0.00
Traffic, 31 sequences						
mean	1.59	**0.02**	0.06	5.43	0.2	0.48
median	1.17	0.00	0.00	1.48	0.00	0.00
Checkerboard, 78 sequences						
mean	1.55	1.12	3.62	2.57	1.69	**0.56**
median	0.29	0.00	0.00	0.27	0.00	0.00
All 120 sequences						
mean	2.40	0.82	2.62	3.45	1.52	**0.53**
median	0.43	0.00	0.00	0.59	0.00	0.00

Table 2. Mean and median of the misclassification (%) on the Hopkins 155 dataset with three motions.

Method	ALC	SSC	MSMC	LSA	Our$_{pca}$	Our$_{spca}$
Articulated, 2 sequences						
mean	21.08	1.91	1.42	7.25	3.72	3.19
median	21.08	1.91	1.42	7.25	3.72	3.19
Traffic, 7 sequences						
mean	7.75	0.58	0.16	25.07	0.19	0.72
median	0.49	0.00	0.00	5.47	0.00	0.19
Checkerboard, 26 sequences						
mean	5.20	2.97	8.30	5.80	5.01	1.22
median	0.67	0.27	0.93	1.77	0.78	0.55
All 35 sequences						
mean	6.69	2.45	3.29	9.73	2.97	1.94
median	0.67	0.20	0.78	2.33	1.50	1.30

Table 3. Mean and median of the misclassification (%) on all the Hopkins 155 dataset.

Method	ALC	SSC	MSMC	LSA	Our$_{pca}$	Our$_{spca}$
all 155 sequences						
Mean	3.56	1.24	2.96	4.94	1.98	0.70
Median	0.50	0.00		0.90	0.75	0.00

Table 4. Computation time (sec) on all the Hopkins 155 dataset.

Method	ALC	SSC	Our$_{pca}$	Our$_{spca}$
Run time [sec.]	88831	14500	1066	1394

Table 1 and Table 2 show that the overall error rate of ours with sparse PCA projection is the lowest for both 2 and 3 motions. Generally, the PCA projection has a lower accuracy than sparse PCA projection for the articulated and checkerboard sequences. However, the traffic video with PCA projection reaches a better result than the sparse PCA projection. The reason is that the PCA projection is more robust to represent the rigid motion, while the sparse PCA projection is more robust to

represent the independent and non-rigid motions. The checkerboard data is the most significant part of the entire Hopkins dataset. It has many intersection problems between different motions. Our framework with sparse PCA projection gives the most accurate results for the checkerboard sequences, for both two and three motions, which means that our method is most accurate for clustering different intersected motions. Table 3 shows that our method achieves the lowest misclassification error for all the sequences from the Hopkins dataset in comparison with all the other algorithms. We evaluate our method with sparse PCA projection in comparison with LSA [11], SSC [6], MSMC [4], GPCA [9], RANSAC [10] and MSMC [4] in Figure 5 and Figure 6 on the Hopkins 155 dataset. Note that MSMC has not been evaluated on the checkerboard sequences.

Freiburg-Berkeley Motion Segmentation Dataset

In this section, our method is evaluated on the Freiburg-Berkeley Motion Segmentation dataset [5] to test the performance on the real video sequences with occlusion and moving camera problems. This dataset contains 59 sequences and all the feature trajectories are tracked densely. All the missing trajectories have not been removed. The parameters for evaluation are precision (%) and recall (%). Our method is compared with SSC [6], ALC [7], and Ochs *et al.* [5], which is based on the affinity of the trajectories between two frames. The results on all the training set and test set of the Berkeley dataset are shown in Table 5.

Table 5. Results on the Entire Freiburg-Berkeley Motion Segmentation Dataset [5]

	Ochs	ALC	SSC	Our$_{pca}$	Our$_{spca}$
Precision	82.36	55.78	64.55	72.12	70.77
Recall	61.66	37.43	33.45	66.52	65.42

As shown in Table 5, the PCA projection outperforms the sparse PCA on this dataset. More specifically, our method with PCA projection obtains the highest recall comparing with the others, which indicates our assigned clusters can cover the most parts of the different ground truth regions. However, compared with Ochs *et al.* [5], which is based on the affinity, our method is lower with respect to the precision. It means that our method can detect the boundaries of different regions correctly, but cannot complete segmenting the moving objects from the background. Figure 7 shows the examples of our results with PCA projection. Our method has high quality segmentation of the primary foreground moving objects. In comparison with SSC and ALC, our method has a superior performance on the precision and the recall. Figure 8 shows some additional segmentation results. The typical failure segmentation is shown in the bottom row *marple1.avi*.

Conclusions

In this paper, we propose a subspace-based framework for segmenting multiple moving objects from a video sequence with the global and local sparse subspace optimization methods. The sparse PCA performs the data projection from a high-dimensional subspace to a global subspace with the sparse orthogonal principal vectors. We seek a sparse representation for the nearest neighbors in the global subspace for the data point spanning a same local subspace. Furthermore, we propose an error estimation function to refine the local subspace estimation for the missing data. The limitation of our work is the number of the motions should be known. We evaluate our proposed framework by comparing it to other motion segmentation algorithms. Our method achieves improved performance on two state-of-the-art benchmark datasets.

Acknowledgments

The work is funded by DFG (German Research Foundation) YA 351/2-1. The authors gratefully acknowledge the support.

References

[1] M.Y. Yang, and B. Rosenhahn, 2014. Video segmentation with joint object and trajectory labeling, *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pp. 831–838.

[2] M.Y. Yang, S.Feng, H. Ackermann, and B. Rosenhahn, 2015. Global and local sparse subspace optimization for motion segmentation, *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, ISA15*.

[3] R. Tron, and R. Vidal, 2007. A benchmark for the comparison of 3-d motion segmentation algorithms, *Computer Vision and Pattern Recognition*, pp. 1–8.

[4] R. Dragon, B. Rosenhahn, and J. Ostermann, 2012. Multi-scale clustering of frame-to-frame correspondences for motion segmentation, *Proceedings of the European Conference on Computer Vision*, pp. 445–458.

[5] P. Ochs, J. Malik, and T. Brox, 2014. Segmentation of moving objects by long term video analysis, *Pattern Analysis and Machine Intelligence*, 36(6):1187–1200.

[6] E. Elhamifar and R. Vidal, 2009. Sparse subspace clustering, *Computer Vision and Pattern Recognition*, pp. 2790–2797.

[7] Y. Ma, H. Derksen, W. Hong, and J. Wright, 2007.Segmentation of multivariate mixed data via lossy data coding and compression, *Pattern Analysis and Machine Intelligence*, 29(9):1546–1562.

[8] H. Zou, T. Hastie, and R. Tibshirani, 2006. Sparse principal component analysis, *Journal of Computational and Graphical Statistics*, 15(2):265–286.

[9] R. Vidal, Y. Ma, and S. Sastry, 2005. Generalized principal component analysis (gpca), *Pattern Analysis and Machine Intelligence*, 27(12):1945–1959.

[10] M.A. Fischler, and R.C. Bolles, 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM*, 24(6):381–395.

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

[12] K. Kanatani, 2001. Motion segmentation by subspace separation and model selection, *Proceedings of the International Conference on Computer Vision*, pp. 586–591.

[13] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre, 2010. Generalized power method for sparse principal component analysis, *The Journal of Machine Learning Research*, 11:517–553.

[14] A. Goh, and R. Vidal, 2007. Segmenting motions of different types by unsupervised manifold clustering, *Computer Vision and Pattern Recognition*, pp. 1–6.

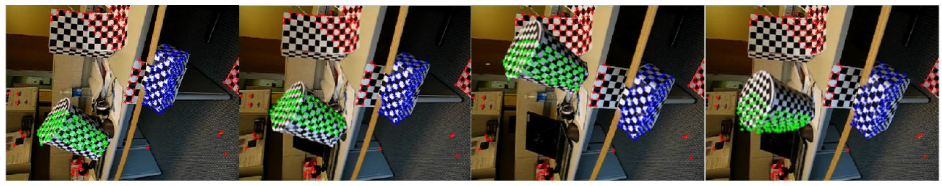
[15] L. Zappella, X. Lladó, E. Provenzi, and J. Salvi, 2011. Enhanced local subspace affinity for feature-based motion segmentation, *Pattern Recognition*, 44(2):454–470.

[16] E. Elhamifar, and R. Vidal, 2011. Sparse manifold clustering and embedding, *Proceedings of Neural Information Processing Systems*, pp. 55–63.

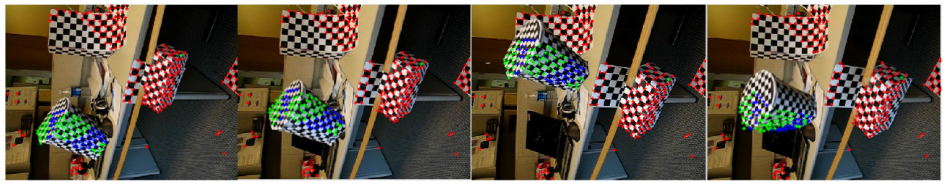
[17] N.P. daSilva and J.P. Costeira, 2009. The normalized subspace inclusion: Robust clustering of motion subspaces, *Proceedings of the International Conference on Computer Vision*, 2009, pp. 1444–1450.

[18] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers, *Foundations and Trends in Machine Learning*, 3(1):1–122.

[19] U. Von Luxburg, A tutorial on spectral clustering, 2007. *Statistics and Computing*,17(4): 395–416.

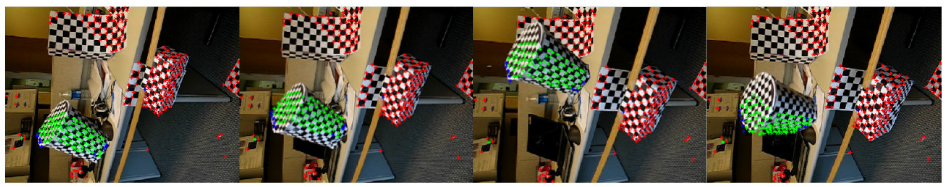


(a)



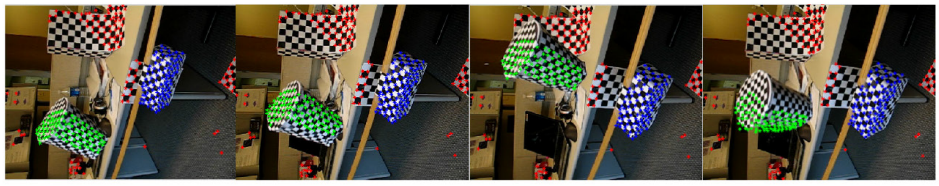
(b)

(c)



(d)

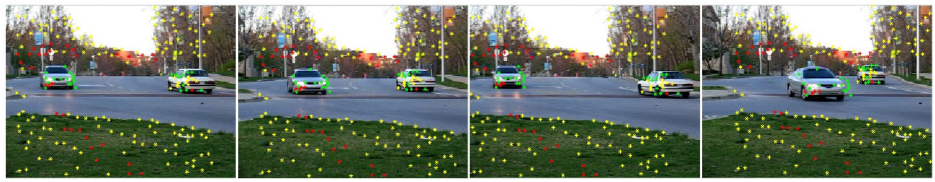
(e)



(f)

Figure 5. Comparison of our method with the ground truth and the other approaches on the *1RT2RC* video: (a): Ground Truth; (b): GPCA, error: 44.98%; (c): LSA, error:1.94%; (d): RANSAC, error: 33.66%; (e): SSC, 0%; (f): Ours, 0% on the *1RT2TC* sequence from the Hopkins 155 dataset.

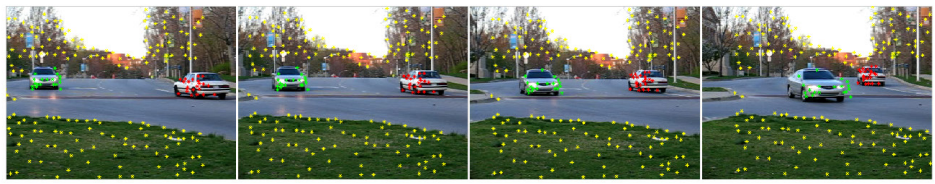
(a)



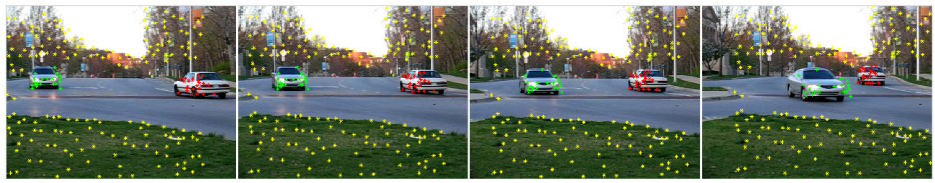
(b)

(c)

(d)

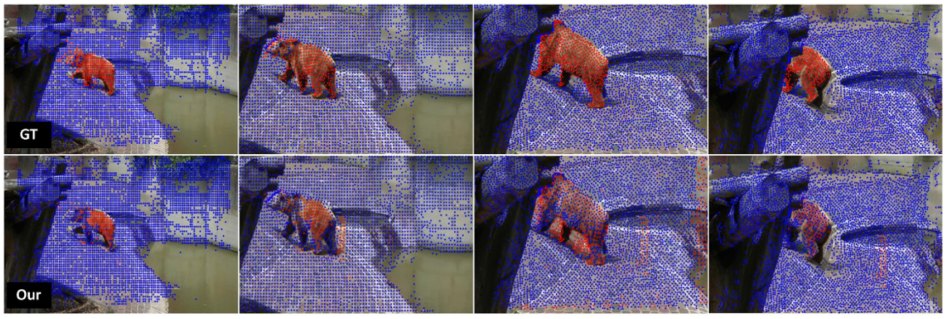


(e)



(f)

Figure 6. Comparison of our method with the ground truth and the other approaches on the *1RT2RC* video: (a): Ground Truth; (b): GPCA, error: 19.34%; (c): LSA, error:46.23%; (d) MSMC, error: 46.23%; (e) SSC, 0%; and (f): Ours, 0%.



(a)

(b)

(c)

Figure 7. Our segmentation results on Freiburg-Berkeley Motion Segmentation Dataset in comparison with the ground truth segmentation [5]: (a) :bear01, (b): marple4, (c): cars8.

Figure 8. Additional segmentation results of Freiburg-Berkeley Motion Segmentation Dataset [5].