# Motion Segmentation via a Sparsity Constraint

Taotao Lai, Hanzi Wang*, *Senior Member, IEEE,*
Yan Yan, *Member, IEEE,* Tat-Jun Chin, *Member, IEEE,* and Wan-Lei Zhao

***Abstract*—Motion segmentation is an important task for intelligent transportation systems. In this paper, inspired by the fact that a feature point trajectory can be sparsely represented as a combination of several feature point trajectories that share coherent transformations, an efficient and effective motion segmentation method with a sparsity constraint is proposed. Specifically, we first propose an accumulated scheme to efficiently integrate motion information from all frames of a video sequence to construct a correlation matrix. Then a sparse affinity matrix is built on the correlation matrix by using information theoretic principles, where the nonzero elements in the same row of the sparse affinity matrix correspond to the feature point trajectories more likely belonging to the same motion. Thereafter, a segment and merge procedure is proposed to effectively estimate the number of motions via the sparse affinity matrix. Finally, by applying spectral clustering on the sparse affinity matrix, different motions in the video sequence are accurately segmented based on the estimated number of motions. Experimental results on the *Hopkins 155* and the *62-clip* datasets demonstrate that the proposed method achieves superior performance compared with several state-of-the-art methods.**

***Index Terms*—Motion segmentation, affinity-based method, residual sorting, sparsity constraint.**

## I. Introduction

Motion segmentation is one of the most important research areas in computer vision. It has been used as a pre-processing step for many applications in intelligent transportation systems, such as visual surveillance, object tracking, action recognition. The aim of motion segmentation is to recognize and separate different moving objects (such as moving vehicles or moving people) according to their different motion patterns, where each moving object is identified as a coherent entity. Numerous motion segmentation methods (e.g., [1]–[9]) have been proposed recently. Many of them have demonstrated excellent performance on popular test benchmarks, e.g., the *Hopkins 155* dataset [10].

According to the recent works [5], [6], motion segmentation methods can be grouped into two-frame based and multi-frame based methods. The latter type has attracted more attention due to its ability to exploit the motion information from all frames of a video sequence for accurate motion segmentation. The authors of [11] roughly group previous multi-frame based motion segmentation methods into two categories: subspace-based methods (e.g., [9], [11]–[16]), and affinity-based methods (such as [5], [6], [11], [17]). Subspace-based methods segment different motions based on a data matrix constructed by using all feature point trajectories of a video sequence. On the other hand, affinity-based methods segment different motions based on an affinity matrix constructed from affinities between pairs of feature point trajectories. The proposed method belongs to affinity-based methods.

Although good performance has been observed on the *Hopkins 155* dataset, subspace-based methods may fail to deal with several problems in practice that do not affect affinity-based methods significantly. For instance, subspace-based methods generally show poor performance when objects are temporarily occluded. In such a case, the feature point trajectories corresponding to the occluded objects are missing. Although the use of matrix completion is able to recover missing data [18], there is no guarantee that missing data can be completely recovered. The experimental results of [6] have shown that, even for these subspace-based motion segmentation methods integrated with the step of missing data recovery, satisfactory performance has not been achieved when some of the feature point trajectories are missing.

In contrast, affinity-based methods suffer less from object occlusions since they only require that feature point trajectories are visible in at least two frames. As indicated in recent work [5], [6], affinity-based methods are able to achieve competitive performance. However, the major disadvantage of the method in [5] is that the number of motions needs to be predefined, which is not practical. Although the method in [6] is able to estimate the number of motions automatically, its computational cost is pretty high. For example, on the video sequence *Van* (in the *62-clip* dataset), which consists of 100 frames and 802 feature points per frame, the method in [6] took 5,237 seconds to obtain the motion segmentation results. Thus, it is difficult to be used in real applications.

Recently, both BM (Brox and Malik [17]) and OB (Ochs and Brox [19]) have been proposed to perform motion segmentation by clustering dense feature point trajectories. However, BM constructs pairwise similarities by restricting motion models to be 2D translations. Therefore, BM may fail if motions undertake complex models [20]. Moreover, a drawback of OB is its high computational cost. For instance, OB spends 48 minutes to compute the affinity matrix of the video sequence *car1* [20], while *car1* consists of only 19 frames and 4,850 feature points per frame.

In this paper, we focus on the more accurate motion model using a lower number of feature point trajectories. We present an efficient and effective Motion Segmentation method with a Sparsity Constraint (called MSSC). The proposed method is based on the observation that a motion can be viewed to be composed of a number of homographies (i.e., in our

Taotao Lai, Hanzi Wang, Yan Yan and Wan-Lei Zhao are with the Fujian Key Laboratory of Sensing and Computing for Smart City, School of Information Science and Engineering, Xiamen University, Xiamen 361005, China (e-mail: laitaotao@gmail.com, hanzi.wang@xmu.edu.cn, yanyan@xmu.edu.cn, wlzhao@xmu.edu.cn).

Tat-Jun Chin is with the ACVT and School of Computer Science, The University of Adelaide, Adelaide, SA 5005, Australia (e-mail: tat-jun.chin@adelaide.edu.au).

*Corresponding author.

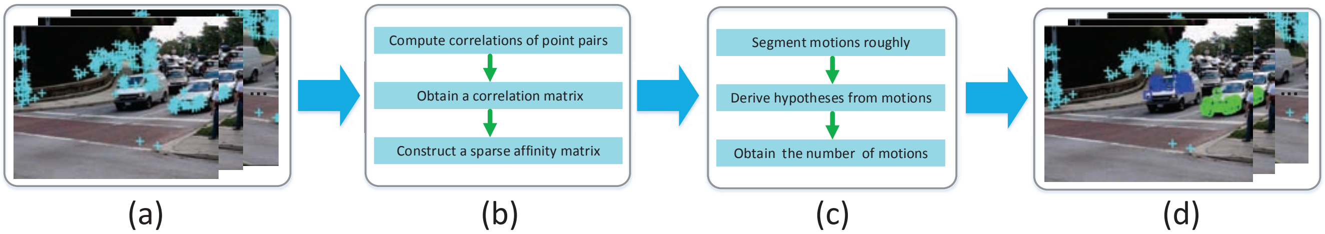

Compute correlations of point pairs
Obtain a correlation matrix
Construct a sparse affinity matrix
Segment motions roughly
Derive hypotheses from motions
Obtain the number of motions
(a)
(b)
(c)
(d)

Fig. 1. Overview of the proposed MSSC method for motion segmentation. (a) The input video sequence *Cars9* of the *Hopkins 155* with the tracked feature points. (b) Based on the input feature points, the proposed MSSC constructs a sparse affinity matrix. (c) MSSC estimates the number of motions in the video sequence via the sparse affinity matrix. (d) Spectral clustering is applied on the sparse affinity matrix to accurately segment motions with the estimated number of motions. Three segmented motions are marked with different colors.

work, we approximate a rigid motion using a finite number of homography matrices, which is sufficient to deal with many objects in real environments, e.g., vehicle, pedestrian). Thus, motion segmentation (which is usually solved by using the fundamental matrix model) can be split into a number of simpler homography matrix estimation tasks. More specifically, we follow the sparse self-expression of the Sparse Subspace Clustering (SSC) [4] due to its good performance. In this paper, a sparse representation of a feature point trajectory corresponds to choosing several other feature point trajectories that share similar homographies.

An overview of the proposed MSSC is shown in Fig. 1. MSSC first constructs a sparse affinity matrix in Fig. 1(b) using the input feature point trajectories (shown in Fig. 1(a)). Based on the constructed sparse affinity matrix, MSSC then estimates the number of motions in Fig. 1(c). Finally, MSSC segments different motions via both the sparse affinity matrix and the estimated number of motions in Fig. 1(d).

It is worth pointing out that the main differences between the proposed MSSC and the recently proposed affinity-based methods [5], [6] are as follows: (1) MSSC can accurately estimate the number of motions from the constructed sparse affinity matrix using Mutual Information Theory (MIT) while the method in [5] needs to specify the number of motions to construct the affinity matrix. (2) Compared with the method in [6], MSSC works within a reasonable time because it solves the motion segmentation problem by splitting the problem to a number of homography matrix estimation tasks[1] while the method in [6] pursuits the result by using a more complex geometric model (i.e., the fundamental matrix model) in a mixed norm optimization scheme, leading to high computational cost.

The main contributions of this paper are summarized as follows. First, we propose a simple but effective accumulated scheme, by which motion information from all frames of a video sequence is effectively integrated into an accumulated correlation matrix. Second, based on the accumulated correlation matrix, we propose to use information theoretic principles to construct a sparse affinity matrix for improving the accuracies of both estimating the number of motions and segmenting different motions. Third, we propose a novel hypothesis generation strategy to accurately estimate the number of motions in the video sequence via the sparse affinity matrix.

The rest of the paper is organized as follows. In Section II, we review the related work. In Section III, we present a new method for constructing a sparse affinity matrix by using both residual sorting and information theoretic principles. In Section IV, we propose an effective scheme for estimating the number of motions by using MIT. In Section V, we show experimental results, and we draw conclusions in Section VI.

## II. Related Work

In this section, we first review the *subspace-based* motion segmentation methods in Section II-A. Then, we introduce the *affinity-based* motion segmentation methods in Section II-B. Finally, we review the methods, which estimate the number of motions for motion segmentation, in Section II-C.

### A. Subspace-based motion segmentation methods

Several subspace-based methods, such as [4], [7], [9], [12]–[16], have been proposed in recent years. These methods can be further classified into three categories: algebraic methods, information-theoretic methods and spectral clustering-based methods.

• Some algebraic methods based on factorization (e.g., [14], [15]) segment different motions by decomposing a matrix composed of feature point trajectories directly. These methods assume that the motions are independent. However, this assumption does not hold across all the cases in practice. Some algebraic-geometric methods (e.g., Generalized Principal Component Analysis (GPCA) [16]) are able to deal with motion subspaces with different dimensions. Nevertheless, these methods are sensitive to noises [4].

• Information-theoretic methods (e.g., Agglomerative Lossy Compression (ALC) [9]) aim to minimize the coding length to represent feature point trajectories. ALC first treats each feature point trajectory as an independent group, and then iteratively merges pairs of groups to maximally reduce the coding length. Note that the proposed method also uses the information theoretic principles. However, different from ALC [9], the proposed method uses the information theoretic principles to adaptively obtain thresholds for constructing sparse affinity matrices.

[1] The running time of generating promising hypotheses significantly reduces when the complexity of the geometric model decreases [21], especially for random sampling.

• Spectral clustering-based methods (such as [4], [7], [12], [13], [22], [23]) firstly construct a similarity matrix between pairs of feature point trajectories based on a data matrix constructed using all feature point trajectories of a video sequence. The spectral clustering technique is then adopted to segment the feature point trajectories into different motions based on the similarity matrix. One advantage of the spectral clustering-based methods is that they are robust to noises, which has been demonstrated by the good performance on the *Hopkins 155* dataset by most of the methods in this category. However, these methods encounter the performance bottleneck when confronting some practical problems (e.g., object occlusions), as pointed out in Section I. Note that although the proposed method is also a spectral clustering-based method, the proposed method is an affinity-based method and thus it suffers less from object occlusions.

### B. Affinity-based motion segmentation methods

In order to segment different motions, several affinity-based motion segmentation methods (e.g., [5], [6]) consider the epipolar constraint. For instance, in [5], the Randomized Voting (RV) score for each feature point is computed based on the distance between the feature point and the corresponding epipolar line. These scores are accumulated to segment motions. Feature point information of each image pair has been exploited in [6], where the feature point information of multiple image frames is further integrated via a mixed norm optimization scheme. However, the drawbacks of [5], [6] are that: the number of motions has to be predefined in [5], which is not realistic in practice and the Multiple-Two-Perspective-View (MTPV) method in [6] is computationally expensive.

The Multi-Scale Motion Clustering (MSMC) [11] method also belongs to affinity-based methods. MSMC iteratively performs the "split-and-merge" procedure to segment different motions. However, it is vulnerable to complicated scenes due to the use of the homography motion model. Note that the proposed method also uses the homography motion model. Nevertheless, we do not construct an affinity matrix based on the inliers of homography matrices as done in [11]. Thus, the proposed method does not require to estimate the scales of inliers. While MSMC needs to predefine the scales of inliers, which is not trivial. In contrast, the proposed method is more robust, since it uses both residual sorting and information theoretic principles to construct a sparse affinity matrix, in which a feature point trajectory correlates to only several feature point trajectories that share similar homographies. Thereby, feature point trajectories from different motions are accurately correlated to each other. Moreover, the strategy of the proposed method for estimating the number of motions is more accurate than that of MSMC. Specifically, the proposed method uses a segment and merge procedure to effectively overcome the problem of over-segmentation. While MSMC compares the results of clustering with various motion numbers, which may lead to over-segmentation.

### C. Estimation of the number of motions

The estimation of the number of motions is one of the most challenging issues in motion segmentation. The Low-Rank Representation (LRR) [7] and the Ordered Residual Kernel (ORK) [12] estimate the number of motions by using a robust scheme to analyze the number of elements in the Laplacian matrix whose eigenvalues are close to zero. Some other methods, such as [6], first over-segment motions by analyzing the eigenvalues of the Laplacian matrix, then they merge the over-segmented motions. However, the eigenvalues of the Laplacian matrix are sensitive to noise, which will greatly affect the accuracies of these methods. The Minimal Basis Facility Location for Subspace Segmentation (MB-FLSS) [24] method uses an affinity propagation-based method to estimate the number of motions. The Branch-and-Bound (BB) [25] method integrates the linear programming optimization technique to estimate the number of motions and it achieves the state-of-the-art performance at 80.00% on the *Hopkins 155*. The proposed method first applies a more robust technique by analyzing the structure of the eigenvectors of the Laplacian matrix (obtained from the constructed sparse affinity matrix) to segment motions, then it merges the over-segmented motions belonging to the same motion with MIT (see Section IV). In this manner, the proposed method can accurately estimate the number of motions and achieves the state-of-the-art performance.

## III. Sparse affinity matrix construction

In this section, we show how the relations between pairs of feature point trajectories are modelled as an accumulated correlation matrix. We also show how to construct a sparse affinity matrix, based on which different motions can be easily segmented. Specifically, after a small number (e.g., 200) of homography matrices are generated for each pair of consecutive frames, an accumulated correlation matrix is computed by residual sorting. Then, the proposed method constructs a sparse affinity matrix by using information theoretic principles, where a feature point trajectory retains the correlations only with several feature point trajectories that share similar homographies.

### A. Correlation computation

Let $\boldsymbol{\mathcal{X}}^f = \{\boldsymbol{x}_1^f, \boldsymbol{x}_2^f, \ldots, \boldsymbol{x}_N^f\}$ be a set of feature correspondences in the $f$th pair of consecutive frames (defined as the $f$th and the $(f+1)$th frames), where $N$ is the number of tracked feature points and $f$ is in the range $[1, F-1]$. Here, $F$ is the number of frames of a video sequence. As in [11], a homography is estimated from a minimal subset of four non-missing feature correspondences by using the Direct Linear Transformation (DLT) [26]. The residuals are computed according to the symmetric transfer error.

Let $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_T\}$ be a putative hypothesis set generated from the $f$th pair of consecutive frames by using random sampling [27], where $T$ is predefined as in [12], [30]. The absolute residual $r_{p,t}^f$ of the $p$th feature correspondence $\boldsymbol{x}_p^f$ at the $f$th pair of consecutive frames with regard to the $t$th hypothesis $\theta_t$ is computed as

$$r_{p,t}^f = R(\boldsymbol{x}_p^f, \theta_t), \tag{1}$$

where the function $R(.)$ computes the residual of $\boldsymbol{x}_p^f$ to $\theta_t$.

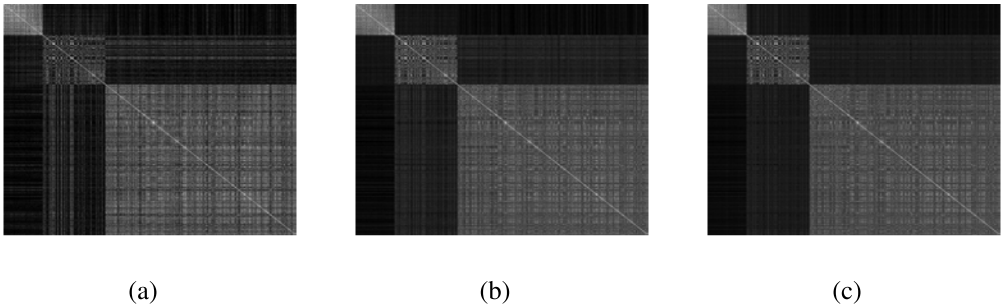


Fig. 2. Correlation matrices computed by the proposed method. (a), (b) and (c) show the correlation matrices computed by (5) on the three-motion video sequence *2T3RCR* of the *Hopkins 155* dataset, accumulating the computed correlations of the first, first several (e.g., eight) and all (twenty five) pairs of consecutive frames, respectively.

For each feature correspondence $\boldsymbol{x}_p^f \in \boldsymbol{\mathcal{X}}^f$, the absolute residual vector $\mathbf{r}_p^f$ between $\boldsymbol{x}_p^f$ and the $T$ generated hypotheses is calculated by (1). $\mathbf{r}_p^f$ is written as

$$\mathbf{r}_p^f = [r_{p,1}^f \; r_{p,2}^f \; \dots \; r_{p,T}^f]. \tag{2}$$

The residuals in $\mathbf{r}_p^f$ are sorted in the non-descending order. Correspondingly, the permutation of residual indices is obtained, i.e.,

$$\boldsymbol{\kappa}_p^f = [\kappa_{p,1}^f \; \kappa_{p,2}^f \; \dots \; \kappa_{p,T}^f]. \tag{3}$$

$\boldsymbol{\kappa}_p^f$ ranks the preference of $\boldsymbol{x}_p^f$ to the $T$ hypotheses. Let $\boldsymbol{\kappa}_{p,1:h}^f$ be the first-$h$ elements of $\boldsymbol{\kappa}_p^f$. As in [21], $h$ is set to $[0.1 * T]$ and the notation $[\cdot]$ means rounding the value. The correlation between two correspondences $\boldsymbol{x}_p^f$ and $\boldsymbol{x}_q^f$ is defined as [21]

$$d_{p,q}^f = \frac{1}{h} \, |\boldsymbol{\kappa}_{p,1:h}^f \cap \boldsymbol{\kappa}_{q,1:h}^f|, \tag{4}$$

where $|\boldsymbol{\kappa}_{p,1:h}^f \cap \boldsymbol{\kappa}_{q,1:h}^f|$ represents the number of identical elements shared by $\boldsymbol{\kappa}_{p,1:h}^f$ and $\boldsymbol{\kappa}_{q,1:h}^f$. The value of $d_{p,q}^f$ is large if $\boldsymbol{x}_p^f$ and $\boldsymbol{x}_q^f$ are from the same *structure* (i.e., motion); otherwise, the value of $d_{p,q}^f$ is low.

In the case that one of feature correspondences is missing due to object occlusions, $d_{p,q}^f$ *is simply set to zero*. There are two reasons for doing this. On one hand, the missing data do not contribute to the correlation of both the $p$th and the $q$th feature correspondences. On the other hand, accumulating the correlations of several pairs of consecutive frames is sufficient to construct the sparse affinity matrix. This will be shown in Section III-B.

Notice that the ORK [12] method constructs an Ordered Residual Kernel (ORK) matrix for motion segmentation by using residual sorting as well. However, the main differences between ORK and the proposed MSSC are threefold: (1) ORK is a subspace-based method while the proposed MSSC is an affinity-based method. Thus, as stated before, ORK is unable to directly handle some practical problems (e.g., object occlusions) while the proposed MSSC can effectively handle these practical problems. (2) ORK segments motions based on the ORK matrix whereas MSSC improves segmentation accuracy by adopting a more robust way: Motion is segmented based on a sparse affinity matrix, which is constructed using both residual sorting and information theory principles. (3) ORK estimates the number of motions by using eigenvalue analysis (see Section II-C) while MSSC presents a more accurate strategy: the number of motions is estimated by using both the eigenvector analysis and MIT (see Section IV).

### *B. Accumulated correlation matrix*

Based on the correlation (defined in (4)) between two feature correspondences from a pair of consecutive frames, an $N \times N$ correlation matrix $\mathbf{D}$ is defined. $D_{i,j}$ is given as

$$D_{i,j} = \sum_{f=1}^{F-1} d^f_{\zeta_f(i),\zeta_f(j)}, \tag{5}$$

where $\zeta_f(i)$ is the index of the correspondence across the $f$th pair of consecutive frames for the feature point trajectory $i$, and similarly for $\zeta_f(j)$. As shown in (5), one entry in matrix $\mathbf{D}$ is an accumulation of $(F-1)$ correlations across all the $F$ frames of a video sequence. The entry $D_{i,j}$ indicates the degree of correlation between the feature point trajectory $i$ and the feature point trajectory $j$ across the whole video sequence. With (5), the correlation between them will be relatively amplified after the accumulation if two feature point trajectories are from the same motion. As observed in Fig. 2(a), the correlation matrix is not accurate enough by computing the correlations of only the first pair of consecutive frames. However, the correlation matrix in Fig. 2(b) accumulates the computed correlations of the first several (e.g., eight) pairs of consecutive frames, which is almost as accurate as accumulating the computed correlations of all (twenty five) pairs of consecutive frames (see Fig. 2(c)).

### *C. Sparse affinity matrix*

The values of the correlations between two feature point trajectories from different motions in $\mathbf{D}$ are usually larger than zero (see Fig. 2(c)). This will affect the subsequent clustering performance. To avoid this problem, we use a similar strategy as the method in [4]: One feature point trajectory is sparsely represented as only several feature point trajectories corresponding to the same motion. As stated in [28], there are two ways to construct a sparse affinity matrix: The $t$-nearest neighbor way and the $\epsilon$-neighborhood way. In the $t$-nearest neighbor way, $D_{i,j}$ is retained if the $j$th (or $i$th) feature point

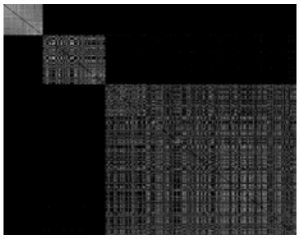


Fig. 3. The constructed sparse affinity matrix by using information theoretic principles based on the correlation matrix of Fig. 2(c).

trajectory is among the $t$ nearest neighbors of the $i$th (or $j$th) feature point trajectory. In the $\epsilon$-neighborhood way, $D_{i,j}$ is set to be zero if the value of $D_{i,j}$ is smaller than a threshold $\epsilon$. In this paper, we use the $\epsilon$-neighborhood way and we use information theory principles to adaptively achieve thresholds for the $\epsilon$-neighborhood way to effectively construct sparse affinity matrices. More specifically, the proposed method first takes advantage of information theory principles to automatically obtain a threshold for the $i$th row of $\mathbf{D}$. Then, the correlations, whose values are smaller than the obtained threshold, are set to be zero for the $i$th row. In this way, a sparse affinity matrix $\mathbf{A}$ is constructed (see Fig. 3), with which the performance of motion segmentation is boosted (see Section V-B). The details are given as follows:

For the $i$th row $\mathbf{D}_i = [D_{i,1}\ D_{i,2} \ldots D_{i,N}]$ of the correlation matrix $\mathbf{D}$, the gap $\delta_j$ between the maximum $\mathbf{D}_i$ and the $j$th element $D_{i,j}$ of $\mathbf{D}_i$ is calculated as:

$$\delta_j = max(\mathbf{D}_i^{\alpha}) - D_{i,j}^{\alpha}. \tag{6}$$

The influence of the parameter $\alpha$ on the performance of the proposed method is studied in Section V-B.

The probability of $\delta_j$ can be written as:

$$\rho(\delta_j) = \delta_j \bigg/ \sum_{k=1}^{N} \delta_k. \tag{7}$$

The entropy of $\mathbf{D}_i$ is obtained by

$$E = \sum_{j=1}^{N} \rho(\delta_j) \log \rho(\delta_j). \tag{8}$$

To this end, $E$ is treated as a threshold to wipe out the entries whose values are smaller than $E$ in the $i$th row, which is given as

$$D_{i,j}^{*} = \left\{ \begin{array}{ll} 0, & if \quad \log \rho(\delta_j) < E \ or \ i = j. \\ D_{i,j}, & otherwise. \end{array} \right. \tag{9}$$

To avoid a trivial solution, the diagonal elements are set to be zero.

After applying the above operations on each row, the correlation matrix $\mathbf{D}$ is updated to form a new matrix $\mathbf{D}^*$. The nonzero entries from the $i$th row of $\mathbf{D}^*$ indicate that the corresponding feature point trajectories are more likely from the same motion as the $i$th feature point trajectory. Finally, a symmetrical sparse affinity matrix $\mathbf{A}$ is defined as

$$\mathbf{A} = \mathbf{D}^* + \mathbf{D}^{*T}, \tag{10}$$

after which a spectral clustering method can be applied to segment motions on $\mathbf{A}$ if the number of motions is predefined. However, the number of motions is usually unknown in practice. The details of how the number of motions is estimated are given in Section IV.

## IV. Estimating the number of motions

In the proposed method, three steps are involved in the estimation of the number of motions. Firstly, by analyzing the eigenvectors derived from the sparse affinity matrix $\mathbf{A}$ which is obtained by (10) (note that the predefined number of motions is not required), the proposed method adaptively segments motions. Since the number of segmented motions may be larger than the real number of motions, merging over-segmented motions is required. Thus, we propose a robust scheme for generating one hypothesis for each segmented motion. Then, the mutual information between each pair of hypotheses, which corresponds to two segmented motions, is computed. Afterwards, a merging step is implemented on each pair of over-segmented hypotheses whose mutual information is larger than zero. After merging, the number of remaining hypotheses is the estimated number of motions.

### A. Segmenting motions

Let $\mathbf{M}$ be a diagonal matrix derived from the affinity matrix $\mathbf{A}$, and $M_{i,i} = \sum_{j=1}^{N} A_{i,j}$. The normalized Laplacian matrix is constructed as $\mathbf{L} = \mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}$. The $C$ eigenvectors of $\mathbf{L}$ are associated with the largest $C$ eigenvalues for $\mathbf{L}$, where $C$ is the largest possible motion number. In this paper, $C$ is fixed to 5 in all experiments following the similar experimental settings as [29]. The $C$ eigenvectors are organized to form the matrix $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_C]$. The matrix $\mathbf{Y}$ is further rotated by $\mathbf{V} = \mathbf{Y}\mathbf{O}$, where $\mathbf{O}$ is the orthogonal matrix which best aligns $\mathbf{Y}$'s columns (see [29]).

Based on $\mathbf{V}$, a cost function defined in [29] is employed:

$$\mathcal{F} = \sum_{n=1}^{N} \sum_{c=1}^{C} V_{i,j}^2 / (\max \mathbf{V}_i)^2. \tag{11}$$

The number of motions $\hat{c}$ which minimizes $\mathcal{F}$ is selected as the estimated number of segments, where $\hat{c} \in \{1, 2, \ldots, C\}$. After this step, the $N$ feature point trajectories are segmented into $\hat{c}$ motions. If there are $\eta$ segmented motions with a small number (e.g., less than 8) of feature correspondences, these feature correspondences cannot adequately represent the corresponding motion. In such cases, the motions with a small number of feature correspondences are deleted, and the estimated number of motions $\hat{c}$ is set to be $(\hat{c} - \eta)$. However, the estimated number $\hat{c}$ may be larger than the real number of motions. In the following subsection, a novel way based on MIT is presented to merge correlated motions.

### B. Deriving hypotheses from segmented motions

In this section, we describe how to derive a hypothesis from a segmented motion. We show the process of deriving the hypothesis for the $m$th segmented motion, where

$m \in \{1, 2, ..., \hat{c}\}$ and $\hat{c}$ is the number of segmented motions obtained by minimizing (11). Specifically, let $\tilde{\boldsymbol{\mathcal{X}}}^{m,f} = \{\boldsymbol{x}_{\lambda_1}^{m,f}, \boldsymbol{x}_{\lambda_2}^{m,f}, \ldots, \boldsymbol{x}_{\lambda_k}^{m,f}\}$ be $k$ feature correspondences of the $m$th segmented motion in the $f$th pair of consecutive frames after swiping missing data. Here, $\{\lambda_1, \lambda_2, \ldots, \lambda_k\} \subset \{1, 2, \ldots, N\}$ is a set of data indices. Based on $\tilde{\boldsymbol{\mathcal{X}}}^{m,f}$, the proposed method first computes a fundamental matrix estimate $\theta_{m,f}$ using the 8-point method [26] if $k \geq 8$. Then, the proposed method calculates the residual $r_{\zeta_f(i),m}^{f}$ of the $\zeta_f(i)$th correspondence with respect to the hypothesis $\theta_{m,f}$ using the Sampson distance [26], if the $\zeta_f(i)$th correspondence is not missing in the $f$th pair of consecutive frames, where $\zeta_f(i)$ is the index of the correspondence across the $f$th pair of consecutive frames for the feature point trajectory $i$. Otherwise, the proposed method does not compute the residual for the $\zeta_f(i)$th correspondence.

In the same way, the proposed method calculates $Q$ $(Q > 0)$ residuals for the $i$th feature point trajectory in all the $(F-1)$ pairs of consecutive frames of a video sequence with $F$ frames,

$$\boldsymbol{r}_{i,m} = [r_{\zeta_{a_1}(i),m}^{a_1} \; r_{\zeta_{a_2}(i),m}^{a_2} \; \cdots \; r_{\zeta_{a_Q}(i),m}^{a_Q}], \tag{12}$$

where $\{a_1, a_2, \ldots, a_Q\} \subseteq \{1, 2, \ldots, F-1\}$ are the indices of the $Q$ pairs of consecutive frames in the $(F-1)$ pairs of consecutive frames if the $i$th feature point trajectory is not missing in the $Q$ pairs of consecutive frames. To make the proposed method robust to noises, the median of the residuals,

$$\hat{r}_{i,m} = \text{median}\{\boldsymbol{r}_{i,m}\}, \tag{13}$$

is selected as the residual of the $i$th feature point trajectory with respect to the hypothesis $\hat{\theta}_m$, which is associated with the $m$th segmented motion. Likewise, the proposed method is able to calculate the residual vector $\hat{\boldsymbol{r}}_m$ of the $N$ feature point trajectories with regard to the $m$th hypothesis $\hat{\theta}_m$, yielding,

$$\hat{\boldsymbol{r}}_m = [\hat{r}_{1,m} \; \hat{r}_{2,m} \; \cdots \; \hat{r}_{N,m}]. \tag{14}$$

In the same way, the proposed method calculates $\hat{c}$ residual vectors for $N$ feature point trajectories against $\hat{c}$ hypotheses, which are associated with $\hat{c}$ segmented motions obtained in Section IV-A.

### C. Obtaining the number of motions by merging hypotheses

After obtaining the $\hat{c}$ residual vectors corresponding to $\hat{c}$ hypotheses, the proposed method computes the mutual information $M(\hat{\theta}_m, \hat{\theta}_n)$ between each pair of hypotheses as in [30], where $m \in \{1, 2, \ldots, \hat{c}\}$ and $n \in \{1, 2, \ldots, \hat{c}\}$. According to mutual information theory, the value of mutual information between two hypotheses is low if they are derived from different motions. On the contrary, two hypotheses hold a large value of mutual information if they are derived from the same motion. Two hypotheses are merged if the mutual information $M(\hat{\theta}_m, \hat{\theta}_n)$ between the two hypotheses is larger than zero. Otherwise, the two hypotheses are kept separate. The details of the mutual information computation can be found in [30]. After merging, the number of the remaining hypotheses is regarded as the estimated number of motions. The effectiveness of the proposed merging scheme on the performance of estimating the number of motions is evaluated in Section V-C.

## V. Experiments

In this section, we evaluate the performance of the proposed MSSC in comparison with several state-of-the-art motion segmentation methods including subspace-based methods (such as SSC [4], LRR [7]) and affinity-based methods (e.g., MTPV [6] and MSMC [11]). In Section V-A, two datasets and evaluation metrics are described. The influence of the parameter $\alpha$ on the performance of the proposed MSSC is evaluated in Section V-B. The motion segmentation results on the two datasets are shown in Section V-C and Section V-D, respectively.

### A. Datasets and evaluation metrics

In the experiments, two datasets are adopted for evaluation, i.e., *Hopkins 155* [10] and *62-clip* [6]. The *Hopkins 155* is one of the most popular benchmarks for motion segmentation. It consists of 120 two-motion video sequences and 35 three-motion video sequences. The *62-clip* dataset is mainly derived from the *Hopkins 155* dataset, including 50 video sequences from the *Hopkins 155*. Another 12 video sequences with object occlusions have been added to the *62-clip* dataset, and the 12 video sequences include 4 video sequences from [31] and another 8 video sequences provided by [6]. Among the 12 video sequences, 9 video sequences have perspective effects. There are 26 two-motion video sequences and 36 three-motion video sequences in the 62-clip dataset.

As in [6], [12], the clustering error (CE) is adopted to measure the segmentation accuracy. It is defined as

$$CE = \frac{\textit{number of misclustered points}}{\textit{total number of points}}. \tag{15}$$

The lower the value of CE is, the better the motion segmentation performance is[2]. In addition to CE, the correct rate (CR)

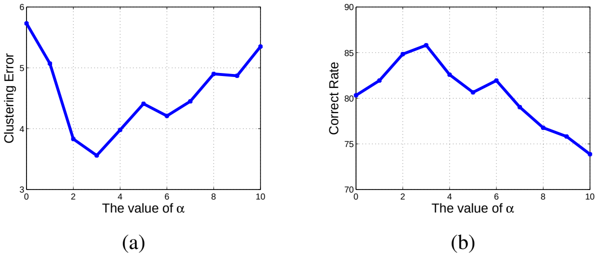
$$CR = \frac{\textit{NVSCE}}{\textit{total number of video sequences}} \tag{16}$$

is also adopted to evaluate the ability of a motion segmentation method to estimate the number of motions in a video sequence, where $NVSCE$ represents the Number of Video Sequences in which the number of motions is Correctly Estimated. The higher the CR is, the better the performance of a method is.

### B. The influence of the parameter $\alpha$

In this section, we study the influence of the parameter $\alpha$ in (6) on the performance of the proposed method. The value of $\alpha$ will affect the sparsity of the matrix obtained by (10): the larger the value of $\alpha$ is, the more sparse the affinity matrix is. We set $\alpha$ in the range of [0 10], where $\alpha = 0$ means that we directly perform clustering on the correlation matrix obtained by (5). For each value of $\alpha$, the proposed method is run 20 times. Fig. 4 shows the median *results* (i.e., the median values of the mean clustering errors and the mean correct rates) obtained by the proposed method over 20 runs on the *Hopkins 155*. From Fig. 4, we can see that the clustering error first decreases and then increases generally

[2] We report the lowest CE by exhaustively evaluating all the cluster pairs using the code provided by the authors of [6] if the estimated number of motions is not equal to the ground truth number.



(a) (b)

Fig. 4. The influence of $\alpha$ on the performance of the proposed MSSC. (a) and (b) show the mean clustering errors (%) and the mean correct rates (%) achieved by the proposed MSSC with different values of $\alpha$, respectively.

whereas the correct rate first increases and then decreases generally as the value of $\alpha$ increases. This is because that, when the value of $\alpha$ is low, the affinity matrix is relatively dense (i.e., the feature point trajectories corresponding to one motion may be correlated with the feature point trajectories of the other motions), leading to under-segmentation. In contrast, when the value of $\alpha$ is large, the feature point trajectories corresponding to one motion may be only correlated with few feature point trajectories from the same motion and thus they cannot effectively represent the corresponding motion, leading to over-segmentation. As a result, the proposed method achieves the lowest clustering error and the largest correct rate when $\alpha = 3$. Therefore, we set the value of $\alpha$ to be 3 in the following experiments.

### C. Results on the *Hopkins 155* dataset

Three experiments are conducted to evaluate the performance of the proposed MSSC method and several other competing methods on the *Hopkins 155* dataset. In the first experiment, the number of motions is predefined and the accuracies of motion segmentation obtained by the proposed MSSC and the other competing methods are evaluated. In the second experiment, the ability of all the competing methods to estimate the number of motions is studied. In the third experiment, the segmentation accuracies obtained by all the competing methods are evaluated in the scenario that the number of motions is unknown. For the proposed MSSC, since the process of sampling is random, each experiment is repeated for 100 times. The median results are reported as the final performance.

In the first experiment, the proposed MSSC is compared with five state-of-the-art methods: ALC [9], SSC [4], LRR [7], Two-Perspective-View (TPV) [6] and RV [5][3]. The results obtained by the five competing methods and the proposed MSSC are given in Table I.

Table I shows that the average clustering errors obtained by the proposed MSSC are less than 2% for video sequences with both two and three motions, which outperform the results achieved by ALC, SSC, LRR and TPV. In contrast to RV, the proposed MSSC outperforms RV on the three-motion video sequences while RV outperforms the proposed MSSC on the two-motion video sequences. On the whole dataset, RV achieves slightly better performance than the proposed MSSC. This is not surprising because RV uses the prior knowledge (i.e., the predefined number of motions) in both affinity matrix construction and spectral clustering while all other methods including the proposed MSSC only use the prior knowledge in spectral clustering. However, the prior knowledge is not available in practice. In contrast, the proposed MSSC is able to estimate the number of motions automatically.

TABLE I
THE CLUSTERING ERROR (%) OBTAINED BY THE SIX COMPETING METHODS ON THE *Hopkins 155* DATASET.

| Method | ALC | SSC | LRR | TPV | RV | MSSC |
|---|---|---|---|---|---|---|
| 2 motions: 120 sequences | | | | | | |
| Mean | 2.40 | 1.52 | 1.33 | 1.57 | 0.44 | 0.54 |
| Median | 0.43 | 0.00 | 0.00 | – | – | 0.00 |
| 3 motions: 35 sequences | | | | | | |
| Mean | 6.69 | 4.40 | 2.51 | 4.98 | 1.88 | 1.84 |
| Median | 0.67 | 0.56 | 0.00 | – | – | 0.30 |
| All | | | | | | |
| Mean | 3.56 | 2.18 | 1.59 | 2.34 | 0.77 | 0.83 |
| Median | 0.50 | 0.00 | 0.00 | – | – | 0.00 |

In the second experiment, the ability of each competing method to estimate the number of motions is evaluated. The performance of the proposed MSSC is studied in comparison with the Ordered Residual Kernel (ORK) [12], the Kernel Optimization (KO) [32], the Low-Rank Representation (LRR) [7], the Minimal Basis(MB)-FLoSS (MB-FLSS) [24], the Branch-and-Bound (BB) [25] and MSSC* (i.e., the proposed MSSC does not use the merging scheme proposed in Section IV while it only uses the *eigenvector* analysis technique (i.e., (11))). The CR results[4] obtained by the competing methods in estimating the number of motions are shown in Table II.

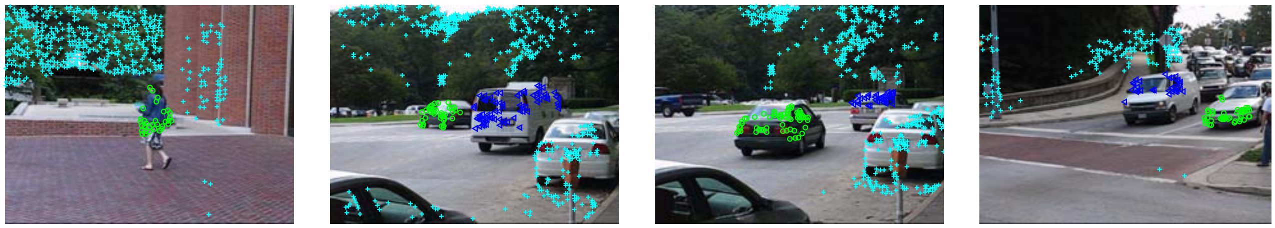
TABLE II
THE CORRECT RATE (%) OF ESTIMATING THE NUMBER OF MOTIONS OBTAINED BY THE COMPETING METHODS ON THE *Hopkins 155* DATASET. THE BEST RESULTS ARE BOLDFACED.

| Method | ORK | KO | LRR | MB-FLSS | BB | MSSC* | MSSC |
|---|---|---|---|---|---|---|---|
| 2 motions | 67.37 | 82.50 | 84.17 | 81.67 | – | 81.67 | **90.00** |
| 3 motions | 49.66 | 48.57 | 57.14 | **71.43** | – | 62.86 | **71.43** |
| All | 63.37 | 74.84 | 78.06 | 79.35 | 80.00 | 77.42 | **85.81** |

As shown in Table II, the proposed MSSC demonstrates good ability in estimating the number of motions in the video sequences. On both the 2-motion and the 3-motion video sequences, the proposed MSSC achieves the highest correct rate compared with the five other competing methods (i.e., ORK, KO, LRR, MB-FLSS and MSSC*). In particular, on the 2-motion video sequences, the correct rate obtained by the proposed MSSC is much higher than those obtained by the five competing methods. Moreover, on the whole video sequences, the correct rate obtained by the proposed MSSC is much higher than the state-of-the-art correct rate (80.00%)

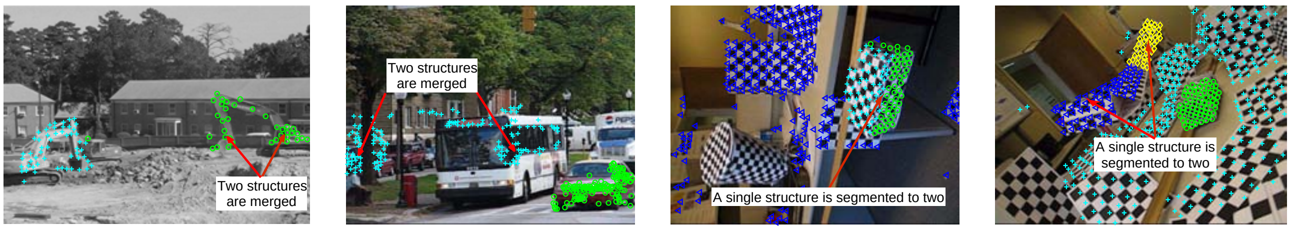
[3] The results of the five competing methods are cited from [4]–[7], [9], respectively. '–' means the value is not reported in the corresponding paper.

[4] The results of KO, LRR and MB-FLSS are cited from [24], while the results of ORK and BB are cited from [12], [25], respectively. '–' means the value is not reported in the corresponding paper.



(a) People1 (b) Cars3 (c) Cars2B (d) Cars9

Fig. 5. Examples showing the successful segmentation cases obtained by the proposed method. (a)-(d) The segmentation results achieved by the proposed method for four video sequences in the *Hopkins 155* dataset.


Two structures are merged
Two structures are merged
A single structure is segmented to two
A single structure is segmented to two

(a) Two_cranes (b) Cars10 (c) 1R2RC_g23 (d) 2T3RCR

Fig. 6. Examples showing the failure cases obtained by the proposed method. (a)-(d) The wrong segmentation results obtained by the proposed method for four video sequences in the *Hopkins 155* dataset.

TABLE III
THE CLUSTERING ERROR (%) OBTAINED BY THE FIVE METHODS ON THE *Hopkins 155* DATASET. THE BEST RESULTS ARE BOLDFACED.

| Method | ORK | LRR | MB-FLSS | BB | MSSC |
|---|---|---|---|---|---|
| 2 motions: 120 sequences | | | | | |
| Mean | 7.83 | 8.59 | 9.45 | – | **2.50** |
| Median | 0.41 | – | – | – | **0.00** |
| 3 motions: 35 sequences | | | | | |
| Mean | 12.62 | 15.51 | 12.07 | – | **7.15** |
| Median | 4.75 | – | – | – | **0.88** |
| All | | | | | |
| Mean | 8.91 | 10.16 | 10.04 | 6.09 | **3.55** |
| Median | – | – | – | **0.00** | **0.00** |

obtained by BB. This is mainly because the proposed MSSC can accurately represent each segmented motion and use MIT to effectively fuse over-segmented motions belonging to the same motion. The correct rate obtained by MSSC* is only 77.42%, which validates the effectiveness of the proposed merging scheme used in MSSC.

In the third experiment, we evaluate the proposed MSSC method in comparison with ORK [12], LRR [7], MB-FLSS [24] and BB [25]. In the experiment, the number of motions is not given to the competing methods and it is to be estimated by the methods[5]. From Table III, we can see that because the proposed MSSC effectively exploits the motion information from all frames of a video sequence, it greatly outperforms the other three competing methods (i.e., ORK, LRR and MB-FLSS) on the video sequences with both two and three motions. On the whole video sequences, the CE obtained by the proposed MSSC is significantly lower than that obtained by the state-of-the-art BB method, which has achieved the second best performance. The differences of the accuracies obtained by the other three competing methods (i.e., ORK, LRR and MB-FLSS) are relatively small, while the gap between the proposed MSSC and the best of the three competing methods is large. Among the three competing methods (i.e., ORK, LRR and MB-FLSS), ORK achieves better performance than LRR and MB-FLSS on both the two-motion video sequences and the whole dataset. On the three-motion video sequences, the clustering error obtained by MB-FLSS is slightly lower than that obtained by ORK. On the whole dataset, MB-FLSS achieves slightly better performance than LRR.

We also show several segmentation results obtained by the proposed MSSC method in Figs. 5 and 6. In Fig. 5, the proposed MSSC successfully segments the four video sequences. In contrast, Fig. 6 shows that MSSC fails in some cases. Figs. 6(a) and 6(b) show the cases where the proposed MSSC fails to estimate the number of motions and in these cases MSSC underestimates the number of motions by (11). Figs. 6(c) and 6(d) show the situations, where a single motion is segmented into two degenerate motions.

### D. Results on the 62-clip dataset

In this section, a thorough evaluation over the proposed MSSC on the *62-clip* dataset is presented. In this experiment, the performance of the proposed method is evaluated in comparison with seven state-of-the-art methods, namely ALC [9], LBF [33], LRR [7], MSMC [11], ORK [12], SSC [4] and MTPV [6]. All the competing methods are required to estimate

[5] The results of LRR and MB-FLSS are cited from [24] whereas the results of ORK and BB are cited from [12], [25], respectively. '–' means the value is not reported in the corresponding paper.

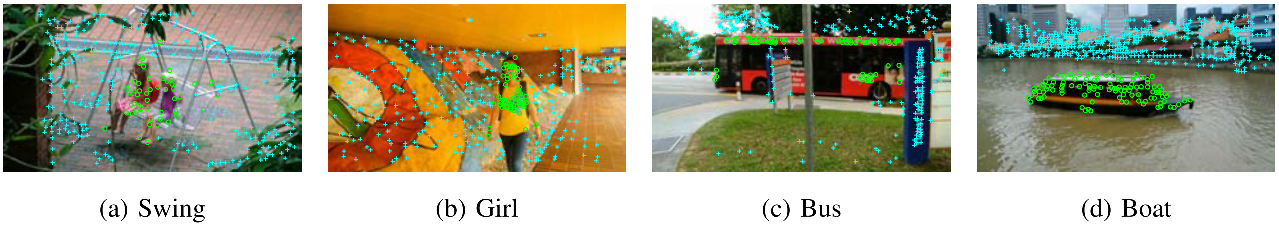

(a) Swing (b) Girl (c) Bus (d) Boat

Fig. 7. The final segmentation results obtained by the proposed method for four video sequences in the *62-clip* dataset. (a)-(d) Examples showing the successful segmentation cases.

TABLE IV
THE CLUSTERING RESULTS OBTAINED BY THE EIGHT COMPETING METHODS ON THE *62-clip* DATASET. THE BEST RESULTS ARE BOLDFACED.

| Method | ALC | LBF | LRR | MSMC | ORK | SSC | MTPV | MSSC |
|---|---|---|---|---|---|---|---|---|
| Clustering error (%) - clips with missing data: 12 clips | | | | | | | | |
| Mean | 25.38(**0.43**) | 20.17(18.47) | 26.03(29.46) | 14.64(1.06) | 24.11(22.33) | 27.41(17.22) | 7.71(0.91) | **1.84**(0.65) |
| Clustering error (%) - clips without missing data: 50 clips | | | | | | | | |
| Mean | 22.03(18.28) | 15.66(1.90) | 9.82(5.26) | 14.19(2.59) | 12.98(4.15) | 13.09(2.01) | 7.56(2.78) | **5.87**(**0.65**) |
| Clustering error (%) - all: 62 clips | | | | | | | | |
| Mean | 22.67(14.88) | 16.53(5.90) | 12.98(5.95) | 14.27(2.34) | 15.13(8.08) | 15.86(5.17) | 7.59(2.37) | **5.09**(**0.65**) |
| Motion number estimation - all 62 clips | | | | | | | | |
| #Correct | 21 | 29 | 35 | 25 | 37 | 33 | 46 | **49** |

the number of motions[6]. For the proposed MSSC, the median results from 100 runs are reported.

In Table IV, the evaluation is divided into three parts. In the first part, the performance of the eight methods on the 12 clips, in which object occlusions exist, is reported. In the second part, the performance of the eight methods on the remaining 50 clips is shown. The third part in the last two rows of Table IV shows the overall performance of the eight methods on the whole *62-clip*. In addition, the clustering error obtained by each method is shown in the bracket by only considering the clips whose motion number is correctly estimated.

As shown in the first row of Table IV (i.e., the first part), we can see that for the challenging 12 clips with missing data, the proposed MSSC achieves the lowest clustering error, because it accurately integrates the motion information from all frames of a video sequence. In contrast, MTPV achieves the second best performance. However, one disadvantage of MTPV is in that its segmentation process is inefficient due to the use of a mixed norm optimization to build the coefficient matrices. For instance, for the *Van* clip with 100 frames, the running time of MTPV is 5,237 seconds, while the proposed MSSC only takes 79 seconds[7]. For ALC, as discussed in [6], the perspective effects have significantly influence on the accuracy of estimating the number of motions, which leads to relatively large error rate. The error rate obtained by ALC on the 12 video sequences is about 13 times higher than that obtained by the proposed MSSC (i.e., 25.38% and 1.84% respectively). The methods such as LBF, ORK, SSC and LRR, which rely on the matrix completion [18] to recover the feature point trajectories of occluded objects, have similar difficulties in handling the structured pattern of the occluded objects. Due to this reason, large error rates are observed for all of these methods. While MSMC fails in complicated clips because it clusters feature correspondences directly based on the inliers of a homography model.

The results of the second part (shown in the second row of Table IV) are for the remaining 50 videos. Although the proposed MSSC achieves the lowest clustering error (i.e., 5.87%), the result can be further boosted by improving the accuracy of motion number estimation. The error rate obtained by the proposed MSSC is only 0.65% when only the clips, whose motion numbers are correctly estimated, are considered. For other competing methods, MTPV achieves the second lowest clustering error (i.e., 7.56%) while all other six competing methods (i.e, ALC, LBF, LRR, MSMC, ORK and SSC) achieve relatively larger clustering errors due to the lower accuracies of estimating the number of motions.

The overall performance of the eight methods is shown in the last two rows of Table IV. The proposed MSSC achieves the lowest clustering error (i.e., 5.09%), which is mainly due to the effective consideration of the motion information from all frames of a video sequence. Moreover, the proposed MSSC is able to correctly estimate the number of motions in 49 out of the 62 video sequences, which is better than the state-of-the-art result achieved by MTPV. For overall performance of the seven other competing methods, MTPV performs second compared to the proposed MSSC whereas all other six competing methods achieve significantly worse performance than both the proposed MSSC and MTPV.

Although the proposed MSSC is much faster than the affinity-based method MTPV, MSSC is slower than the

[6] The results of the seven competing methods are cited from [6]. The methods have all been optimized to achieve their best performance in [6].

[7] The results are provided by the first author of [6] who tested on a laptop with a 1.73GHz Intel i7 Q740 CPU. The first author of [6] runs only one video sequence for us due to expensive computational cost and we cannot obtain more comparative results because the code of the method in [6] is not available. We note that the computational cost of the proposed method is mainly spent on the computation of frame-to-frame correlations (by (4)), which can be easily parallelized to increase computational efficiency.

subspace-based methods (e.g., SSC, LRR). This is because the subspace-based methods pursue segmentation results based on a data matrix constructed using all feature point trajectories of a video sequence. However, the affinity-based methods MTPV and MSSC perform motion segmentation by integrating frame-to-frame motion information from all frames of a video sequence, to more effectively handle practical problem (e.g., object occlusions).

The final segmentation results obtained by the proposed MSSC on four example video sequences of the *62-clip* dataset are shown in Fig. 7, from which we can see that the four video sequences are successfully segmented.

## VI. Conclusions

In this paper, we have presented a novel motion segmentation method (called MSSC), which is able to effectively deal with practical challenges in motion segmentation such as unknown number of motions and object occlusions. MSSC is built upon the accumulation of frame-to-frame affinities into a global affinity. More specifically, MSSC first integrates motion information from all frames of a video sequence into a correlation matrix. Then, MSSC constructs a sparse affinity matrix based on both the correlation matrix and information theoretic principles to boost the performance. After that, MSS-C uses a segment and merge procedure to estimate the number of motions on the constructed sparse affinity matrix. Finally, spectral clustering is applied on the sparse affinity matrix to accurately segment motions with the estimated number of motions. In the experiments, we evaluate the proposed MSSC on two comprehensive datasets and we compare it with several state-of-the-art motion segmentation methods. The promising results demonstrate both the efficiency and effectiveness of the proposed MSSC for motion segmentation.

## Acknowledgment

The authors would like to thank Zhuwen Li for his useful help. This work was supported by the National Natural Science Foundation of China under Grants 61472334, 61571379 and 61572408. Tat-Jun Chin acknowledged funding under ARC DP160103490.

## References

[1] N. K. Kanhere and S. T. Birchfield, “Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features,” *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 148–160, 2008.

[2] D. Gangodkar, P. Kumar, and A. Mittal, “Robust segmentation of moving vehicles under complex outdoor conditions,” *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1738–1752, 2012.

[3] L. Wang and N. H. Yung, “Extraction of moving objects from their background based on multiple adaptive thresholds and boundary evaluation,” *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 40–51, 2010.

[4] E. Elhamifar and R. Vidal, “Sparse subspace clustering: Algorithm, theory, and applications,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, 2013.

[5] H. Jung, J. Ju, and J. Kim, “Rigid motion segmentation using randomized voting,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 1210–1217.

[6] Z. Li, J. Guo, L.-F. Cheong, and S. Z. Zhou, “Perspective motion segmentation via collaborative clustering,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1369–1376.

[7] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, “Robust recovery of subspace structures by low-rank representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, 2013.

[8] B. Poling and G. Lerman, “A new approach to two-view motion segmentation using global dimension minimization,” *Int. J. Comput. Vis.*, vol. 108, no. 3, pp. 165–185, 2014.

[9] S. Rao, R. Tron, R. Vidal, and Y. Ma, “Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 10, pp. 1832–1845, 2010.

[10] R. Tron and R. Vidal, “A benchmark for the comparison of 3-D motion segmentation algorithms,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.

[11] R. Dragon, B. Rosenhahn, and J. Ostermann, “Multi-scale clustering of frame-to-frame correspondences for motion segmentation,” in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 445–458.

[12] T.-J. Chin, H. Wang, and D. Suter, “The ordered residual kernel for robust motion subspace clustering,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 333–341.

[13] J. Feng, Z. Lin, H. Xu, and S. Yan, “Robust subspace segmentation with block-diagonal prior,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 3818–3825.

[14] K. Kanatani, “Motion segmentation by subspace separation and model selection,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2001, pp. 586–591.

[15] C. Tomasi and T. Kanade, “Shape and motion from image streams under orthography: A factorization method,” *Int. J. Comput. Vis.*, vol. 9, no. 2, pp. 137–154, 1992.

[16] R. Vidal, Y. Ma, and S. Sastry, “Generalized principal component analysis (gpca),” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1945–1959, 2005.

[17] T. Brox and J. Malik, “Object segmentation by long term analysis of point trajectories,” in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 282–295.

[18] P. Chen, “Optimization algorithms on subspaces: Revisiting missing data problem in low-rank matrix,” *Int. J. Comput. Vis.*, vol. 80, no. 1, pp. 125–142, 2008.

[19] P. Ochs and T. Brox, “Higher order motion models and spectral clustering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 614–621.

[20] P. Purkait, T.-J. Chin, H. Ackermann, and D. Suter, “Clustering with hypergraphs: The case for large hyperedges,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 672–687.

[21] T.-J. Chin, J. Yu, and D. Suter, “Accelerated hypothesis generation for multistructure data via preference analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 625–638, 2012.

[22] C.-G. Li and R. Vidal, “Structured sparse subspace clustering: A unified optimization framework,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 277–285.

[23] Y. Yan, C. Shen, and H. Wang, “Efficient semidefinite spectral clustering via lagrange duality,” *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3522–3534, 2014.

[24] C.-M. Lee and L.-F. Cheong, “Minimal basis facility location for subspace segmentation,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1585–1592.

[25] H. Hu, J. Feng, and J. Zhou, “Exploiting unsupervised and supervised constraints for subspace clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 8, pp. 1542–1557, 2015.

[26] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision (2nd ed)*. Cambridge University Press, 2004.

[27] M. Fischler and R. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[28] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, and E. Y. Chang, “Parallel spectral clustering in distributed systems,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 568–586, 2011.

[29] L. Zelnik-Manor and P. Perona, “Self-tuning spectral clustering,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 1601–1608.

[30] H. Wang, T.-J. Chin, and D. Suter, “Simultaneously fitting and segmenting multiple-structure data with outliers,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1177–1192, 2012.

[31] K. Schindler, U. James, and H. Wang, “Perspective n-view multibody structure-and-motion through model selection,” in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 606–619.

[32] T.-J. Chin, D. Suter, and H. Wang, “Multi-structure model selection via kernel optimisation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 3586–3593.

[33] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, “Hybrid linear modeling via local best-fit flats,” *Int. J. Comput. Vis.*, vol. 100, no. 3, pp. 217–240, 2012.

**Taotao Lai** received the M.S. degree in Computer Science and Technology from the School of Information Science and Engineering at Xiamen University, Xiamen, China, in 2009. He is currently working toward the Ph.D. degree in Computer Science and Technology at the same University. His research interests include structure from motion and robust model fitting.

**Hanzi Wang** is currently a Distinguished Professor of Minjiang Scholars in Fujian province and a Founding Director of the Center for Pattern Analysis and Machine Intelligence (CPAMI) at Xiamen University in China. He was an Adjunct Professor (2010-2012) and a Senior Research Fellow (2008-2010) at the University of Adelaide, Australia; an Assistant Research Scientist (2007-2008) and a Postdoctoral Fellow (2006-2007) at the Johns Hopkins University; and a Research Fellow at Monash University, Australia (2004-2006). He received his Ph.D degree in Computer Vision from Monash University where he was awarded the Douglas Lampard Electrical Engineering Research Prize and Medal for the best PhD thesis in the Department. His research interests are concentrated on computer vision and pattern recognition including visual tracking, robust statistics, object detection, video segmentation, model fitting, optical flow calculation, 3D structure from motion, image segmentation and related fields.

He is a senior member of the IEEE. He was an Associate Editor for IEEE Transactions on Circuits and Systems for Video Technology (T-CSVT) from 2010 to 2015 and a Guest Editor of Pattern Recognition Letters (September 2009). He was the General Chair for ICIMCS2014, Program Chair for CVRS2012, Publicity Chair for IEEE NAS2012, and Area Chair for DICTA2010. He also served on the program committee (PC) of ICCV, ECCV, CVPR, ACCV, PAKDD, ICIG, ADMA, CISP, etc, and he serves on the reviewer panel for more than 40 journals and conferences.

**Yan Yan** is currently an associate professor in the School of Information Science and Engineering at Xiamen University, China. He received the Ph.D. degree in Information and Communication Engineering from Tsinghua University, China, in 2009. He worked at Nokia Japan R&D center as a research engineer (2009-2010) and Panasonic Singapore Lab as a project leader (2011). He has published around 30 papers in the international journals and conferences including the IEEE T-IP, IEEE T-ITS, PR, KBS, ICCV, ECCV, ACM MM, ICPR, ICIP, etc. His research interests include computer vision and pattern recognition.

**Tat-Jun Chin** received the B.Eng. degree in mechatronics engineering from Universiti Teknologi Malaysia (UTM) in 2003 and the PhD degree in computer systems engineering from Monash University, Victoria, Australia, in 2007. He was a research fellow at the Institute for Infocomm Research (I2R) in Singapore from 2007 to 2008. Since 2008, he has been a lecturer at The University of Adelaide, South Australia. His research interests include robust estimation and statistical learning. He is a member of the IEEE.

**Wan-Lei Zhao** is currently an associate professor in the School of Information Science and Engineering at Xiamen University, China. He received the B.Eng. and M.Eng. degrees from the Department of Computer Science and Engineering, Yunnan University, Kunming, China, in 2006 and 2002, respectively, and the Ph.D. degree from the City University of Hong Kong, Kowloon, Hong Kong, in 2010. He was with the Software Institute, Chinese Academy of Science, Beijing, China, from 2003 to 2004, as an Exchange Student. He was with the University of Kaiserslautern, Kaiserslautern, Germany, in 2011. His current research interests include multimedia information retrieval and video processing.