Perspective Motion Segmentation via Collaborative Clustering

Zhuwen Li[1], Jiaming Guo[1], Loong-Fah Cheong[1] and Steven Zhiying Zhou[1,2]
[1]Dept. of Electrical & Computer Engineering, National University of Singapore
[2]National University of Singapore (Suzhou) Research Institute
{lizhuwen, guo.jiaming, eleclf, elezzy}@nus.edu.sg

Abstract

This paper addresses real-world challenges in the motion segmentation problem, including perspective effects, missing data, and unknown number of motions. It first formulates the 3-D motion segmentation from two perspective views as a subspace clustering problem, utilizing the epipolar constraint of an image pair. It then combines the point correspondence information across multiple image frames via a collaborative clustering step, in which tight integration is achieved via a mixed norm optimization scheme. For model selection, we propose an over-segment and merge approach, where the merging step is based on the property of the ℓ_1-norm of the mutual sparse representation of two oversegmented groups. The resulting algorithm can deal with incomplete trajectories and perspective effects substantially better than state-of-the-art two-frame and multi-frame methods. Experiments on a 62-clip dataset show the significant superiority of the proposed idea in both segmentation accuracy and model selection.

1. Introduction

Previous approaches to the 3D motion segmentation problem can be roughly separated into the multi-frame and the two-frame methods. Multi-frame methods have been studied mostly under the affine assumption, because under this assumption the trajectories of a rigid motion across multiple frames lie in an affine subspace with a dimension of no more than 3, or a linear subspace with a dimension of at most 4. One can then solve the problem using either a factorization or a subspace separation framework [2, 7, 9, 10, 11, 14, 19, 22, 27, 28, 31, 34]. Two-view methods are usually based on the epipolar geometry, and are thus capable of handling perspective effects. The motion model fitting and selection are carried out by either statistical methods [13, 16, 24, 29] or algebraic methods [23, 32, 33].

The multi-frame methods have been better developed, partly due to the elegance of its formulation and partly due to the release of the *Hopkins155* database [30], which contains largely clips with little perspective effects. However, we argue that the current crop of multi-frame affine methods does not confront several real world issues, despite ever-decreasing and near perfect classification rate on *Hopkins155*. There are three major drawbacks of the multi-frame affine methods when compared to the two-frame methods.

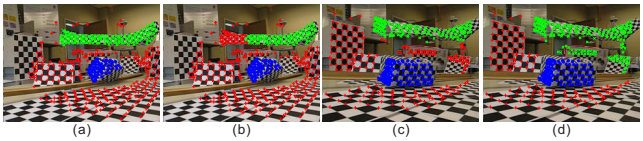

Figure 1. Motion segmentation results of two sequences with strong perspective effects using SSC. The ground truths are shown in (a) and (c), and the SSC results in (b) and (d) respectively.

Firstly, multi-frame affine methods suffer from their inability to deal with perspective effects, while this presents no problem in the two-frame method; it becomes a significant consideration when using shorter lenses for shooting outdoor sequences. Figure 1 shows the results of two sequences with perspective effects from *Hopkins155*; these results are produced by the state-of-the-art clustering algorithm – sparse subspace clustering (SSC) [9]. Compared to the near zero errors achieved by SSC for the other sequences in *Hopkins155* without strong perspective effects, the erroneous segmentation results in these clips are especially notable: in Figure 1(b), part of the green object is classified as belonging to the background, and in Figure 1(d) the green object captures some of the background points.

Secondly, multi-frame affine methods generally require the trajectories to have full-length. If one simply filters out the trajectories which are absent in some frames, the density of the trajectories is likely to be significantly decreased, resulting in lack of coverage of many parts of the sequence. The full-length requirement also makes it difficult to deal with objects entering into or departing from the scene and suffering from temporary occlusion. Figure 2(a) shows the feature points of the "delivery van" data with the full-length requirement on the trajectories. It is observed that they are much sparser than the density of those in Figure 2(b), which only requires the trajectories to appear in at least two frames. Clearly, two-frame methods suffer to a much lesser extent from the missing entry issue. One may

1550-5499/13 $31.00 © 2013 IEEE
DOI 10.1109/ICCV.2013.173

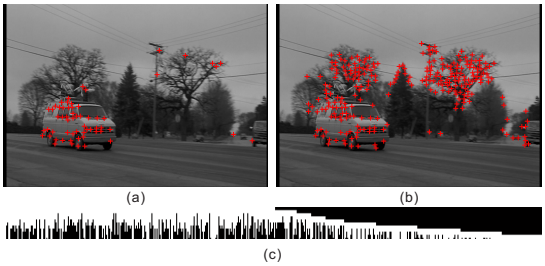

Figure 2. (a) 60 trajectories obtained with the full-length requirement, and (b) 524 trajectories without the full-length requirement. (c) The data matrix, with black area indicating missing entries.

argue that matrix completion techniques can help to fill in the missing entries [5]. However, Candès and Tao [4] have proven a lower bound on the necessary number of uniformly distributed samples, below which no algorithm can guarantee correct recovering of the missing entries. Unfortunately, motion segmentation data often violate this condition. Figure 2(c) shows the data matrix of the "delivery van" data, which has about 50% missing entries and is non-uniformly distributed. Even it is by no means the most challenging data, it is difficult to recover the missing entries.

Thirdly, the number of motion groups is usually assumed to be known *a priori* for multi-frame affine methods. It is indeed a strong indication that model selection is actually difficult for motion segmentation. Related to this issue is the fact that the number of motion groups in each clip of the *Hopkins155* dataset remains unchanged throughout the frames, which makes it easy to indulge in the aforementioned assumption. In real videos, the number of motion groups may change throughout a clip as moving objects enter or leave the scene. Without coming to grips with this fundamental issue, the application of these works to real life problems will be severely hampered. By comparison, the two-frame methods are much better-placed to estimate exactly when moving objects enter or leave the scene.

Despite the relative merits of the two-frame methods over the multi-frame affine methods, less effort is devoted to the two-frame approach in recent years. On the one hand, it is partly due to the belief that multiple frames contain much more information that should be exploited. Contrary to such belief, we will show in Section 4 that the performance of the two-frame method is generally quite adequate; we may indeed question the wisdom of abandoning the two-frame method too hastily, especially in view of the information we lost through these feature points discarded because of the full-length requirement. On the other hand, there are clearly scenes where an observation period as short as two frames may confound the two-frame approach. For example, two objects may be moving with the same motion for a short while but diverge thereafter. In this paper, we propose a multi-frame approach that is rooted in two-frame analysis, with a mixed norm formulation that couples the multi-frame information in an integrated manner. Beginning with a single image pair, we revisit the epipolar constraint of two-perspective-view (TPV), leading to a subspace segmentation problem formulation that segments the null spaces of the appropriate equations. Thus, the idea of subspace separation applies and one can follow the SSC approach in converting the motion segmentation problem into a graph partitioning problem based on an affinity matrix. We prefer the sparse self-expression affinity of SSC, because of its good performance and some degree of tolerance to dependent subspaces [26]. A more powerful formulation that integrates multiple frames then follows, in which we derive an aggregated affinity matrix from multiple image pairs and seek a joint sparse coefficient recovery across multiple image pairs, *i.e.*, the sparse affinity coefficients of a particular trajectory should be consistently distributed across multiple image pairs in the sense that this trajectory should use the same set of other trajectories to express itself across all image pairs. This is formulated as a constrained mixed norm minimization problem, whose relaxed version is convex and can be solved efficiently with augmented Lagrange multiplier (ALM) [18] method.

Another important contribution of our paper lies in its robust model selection scheme. We first make a rough model estimation by analyzing the Laplacian matrix of the affinity matrix and over-segment the data into groups. Then we perform merging by a scheme that takes advantage of the loose grouping already available. Specifically, we use the data points in one group to sparsely represent each data point in another group. Based on Soltanolkotabi and Candès' scheme of outlier rejection [26], which declares a data point to be an outlier if the ℓ_1-norm of its sparse coding vector is above a fixed threshold, we can decide which data points in the second group are inliers w.r.t. the first group and which are outliers. Based on the statistics of the ℓ_1-norm, they can be merged or left as they are.

When evaluated over a 62-clip dataset containing real challenges such as missing data, unknown number of motions, and perspective effects, the results show that our joint inference scheme can produce significantly more accurate and reliable results than those methods individually estimating two-view motion models, followed by a loosely-coupled fusion step, or those state-of-the-art multi-frame methods such as SSC and LRR (low-rank representation [19]).

1.1. Related work

There have been a plethora of multi-frame approaches [2, 7, 9, 10, 11, 14, 19, 22, 27, 28, 31, 34]. While many of them perform very well with *Hopkins155*, significant problems remain, as reviewed in the preceding paragraphs. Our key concern here is to tackle these challenges not well represented in *Hopkins155*. In contrast to the aforementioned approaches, our modelling of the problem is based on the epipolar constraint and does not make concession in terms

of the camera projection, and its multi-frame extension does not suffer from the restriction of requiring features to be present in all frames. While projective factorization [17] extends the camera model to perspective, it needs an iterative process that alternates between the estimation of the depths and the segmentation of the trajectories. Furthermore, it still requires full-length trajectories, and the depth estimation is highly dependent on the initial segmentation.

Two-frame methods [13, 16, 23, 24, 29, 32, 33] are based on the epipolar constraint. Our work is based on the same constraint, though we do not explicitly estimate the fundamental matrices but directly cluster the correspondences. More importantly, our formulation allows multi-frame extension in an integrated manner and can handle incomplete and ambiguous features in a natural way. Thus, compared to works like [8, 25] which are also based on the two-view constraint but extend to multiple frames in a loosely coupled way, our method tightly integrates information from all frames by a global optimization scheme, and is thus expected to achieve more optimal solutions.

Model selection remains very much an open problem in motion segmentation. While the number of zero eigenvalues of the Laplacian matrix can be related to the number of connected components of the affinity matrix, the challenge lies in determining the number of eigenvalues close to zero in a robust manner [19, 26]. Some other methods [6, 8, 13, 16, 25, 24, 29] explicitly generate motion hypotheses and balance the goodness of fit against the complexity of the model. In general, the hypothesis generation step is crucial in determining its success. Models with a high number of parameters face the predicament of generating a sufficiently large number of hypotheses while coping with the prohibitive computational cost. Bad samplings often result in failure for these methods, with the results varying each time due to the sampling procedure. Moreover, it is difficult, probabilistically speaking, to sample an all-inlier minimal set when estimating a high order model, because the number of samples required by the minimal set is relatively larger. Thus, [13, 25] uses calibrated cameras and [8] uses homography, both to reduce the number of points necessary to estimate a motion. For the same purpose, [16, 24] design guided sampling steps. Our method eschews this costly hypothesis generation step but instead takes advantage of the over-segmented grouping provided by the spectral clustering. We then leverage on the recent theoretical result [26] which provides a principled way to detect outlier points based on the ℓ_1 norm of the sparse representation of the point. This in turn allows us to perform merging of two over-segmented groups in a very robust way.

Lastly, some recent research addresses the need to obtain a denser set of trajectories [3, 15]. These works aim to cover the image domain without too many large gaps. However, they only carry out the segmentation in the 2D domain, mainly due to computational consideration. Thus, motions that deviate from the simple 2D model may lead to a wrong segmentation. Our work pays the price of a lower trajectory density for a more accurate motion model and a higher quality data input.

The rest of this paper is organized as follows. Section 2 discusses the TPV subspace in detail. Section 3 describes the joint clustering algorithm and the ℓ_1-norm based merging scheme. Then, our experimental results are illustrated in Section 4. Finally, we draw the conclusion in Section 5.

2. The TPV Motion Subspace

Assume $\mathbf{x}_p = (x_p, y_p, 1)^T$ and $\mathbf{x}'_p = (x'_p, y'_p, 1)^T$ are the homogeneous coordinates of two corresponding points of a 3-D point p in two frames. Their relationship is governed by the epipolar constraint [12] expressed as follows:

$$\mathbf{x}'^{T}_p \mathbf{F} \mathbf{x}_p = 0, \tag{1}$$

where $\mathbf{F} \doteq \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \in \mathbb{R}^{3\times3}$ is the fundamental matrix, which connects correspondences under the same rigid motion in two views. A classic algorithm to compute $\mathbf{F}$ is the 8-point algorithm [12], in which each correspondence gives rise to one linear equation in the unknown entries of $\mathbf{F}$ as follows:

$$(\ x'_p x_p \ \ x'_p y_p \ \ x'_p \ \ y'_p x_p \ \ y'_p y_p \ \ y'_p \ \ x_p \ \ y_p \ \ 1\ )\mathbf{f} = 0, \tag{2}$$

where $\mathbf{f} = (\ f_{11} \ \ f_{12} \ \ f_{13} \ \ f_{21} \ \ f_{22} \ \ f_{23} \ \ f_{31} \ \ f_{32} \ \ f_{33}\ )^T$ is the 9×1 vector made up of the entries of $\mathbf{F}$ in row-major order. The coefficients of this equation are arranged in a column vector, denoted as $\mathbf{w}_p$. Clearly, those $\mathbf{w}_p$ under the same rigid motion k form a hyperplane perpendicular to $\mathbf{f}_k$, which we refer to as the TPV motion subspace. Since $\mathbf{f}_k$ is a 9×1 vector, the dimension of this subspace is at most 8.

A fundamental matrix determines the relationship of a camera pair uniquely [12]. Thus, in general the set of $\mathbf{w}_p$ for points undergoing the same rigid motion k forms a unique hyperplane perpendicular to $\mathbf{f}_k$. However, for points in special configuration, they fail to uniquely determine the fundamental matrix. These include correspondences lying on a plane in space or those only related by a pure rotation about the camera center. In both cases, point correspondences are related by a homography matrix $\mathbf{H} \doteq \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \in \mathbb{R}^{3\times3}$, *i.e.*,

$$[\mathbf{x}'_p]_\times \mathbf{H} \mathbf{x}_p = 0, \tag{3}$$

where $[\mathbf{x}]_\times \in \mathbb{R}^{3\times3}$ denotes the skew-symmetric matrix associated with $\mathbf{x}$. From equation (3), it can be shown that

$\mathbf{w}_p$ is related to a 9×3 matrix $\mathbf{H}'$:

$$\mathbf{w}_p^T \mathbf{H}' = \mathbf{0}, \tag{4}$$

where

$\mathbf{H}' = [\ \mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3\ ]$

$\mathbf{h}_1 = (\ 0\ \ 0\ \ 0\ \ h_{31}\ \ h_{32}\ \ h_{33}\ \ -h_{21}\ \ -h_{22}\ \ -h_{23}\ )^T$,

$\mathbf{h}_2 = (\ -h_{31}\ \ -h_{32}\ \ -h_{33}\ \ 0\ \ 0\ \ 0\ \ h_{11}\ \ h_{12}\ \ h_{13}\ )^T$,

$\mathbf{h}_3 = (\ h_{21}\ \ h_{22}\ \ h_{23}\ \ -h_{11}\ \ -h_{12}\ \ -h_{13}\ \ 0\ \ 0\ \ 0\ )^T$.

It can be observed from (4) that those $\mathbf{w}_p$ under the aforementioned degenerate configurations fall on the intersection of three hyperplanes, each of which is perpendicular to one column of $\mathbf{H}'$. Here, each column of $\mathbf{H}'$ is independent of one another in general and thus the rank of $\mathbf{H}'$ is 3. Thus, $\mathbf{w}_p$ under these degenerate configurations live in a lower dimensional subspace with dimension no more than 6. Fortunately, there are various subspace separation algorithms [9, 19] that can handle subspaces with different dimensions and the above situation should pose no special problem.

3. Clustering Motion Subspaces

3.1. Sparse subspace clustering

The preceding section has reduced the motion segmentation task to that of clustering subspaces of dimension at most 8 in $\mathbb{R}^9$ in general. The data are now collected in a data matrix $\mathbf{W} = [\mathbf{w}_1 \cdots \mathbf{w}_P]$. The SSC algorithm can be used directly to perform subspace clustering for the case of single image pair; the case of multiple image pairs requires joint sparsity and will be discussed in Section 3.1.2.

3.1.1 Single image pair

We briefly review the SSC algorithm in the context of the TPV motion subspace: each column $\mathbf{w}_p$ can be represented as a linear combination of the other columns $\mathbf{w}_q$

$$\mathbf{w}_p = \sum_{q=1, q\neq p}^{P} c_q \mathbf{w}_q = \mathbf{W}_{\widehat{p}} \mathbf{c}_p, \tag{5}$$

where P is the number of correspondences, $\mathbf{W}_{\widehat{p}} = [\mathbf{w}_1 \cdots \mathbf{w}_{p-1}\ \mathbf{w}_{p+1} \cdots \mathbf{w}_P] \in \mathbb{R}^{D \times P-1}$ is the matrix obtained from $\mathbf{W}$ by removing its p-th column and $\mathbf{c}_p \in \mathbb{R}^{P-1}$ is the vector made up of the coefficients c_q. Generally, the solution for (5) is not unique and the key idea of SSC is to obtain a sparsest solution for $\mathbf{c}_p$ via solving the following relaxed ℓ_1 optimization problem

$$\min \|\mathbf{c}_p\|_1 \quad \text{s.t.} \quad \mathbf{w}_p = \mathbf{W}_{\widehat{p}} \mathbf{c}_p. \tag{6}$$

The nonzero entries in the optimal solution $\mathbf{c}_p$ indicate that the corresponding trajectories in $\mathbf{W}_{\widehat{p}}$ belong to the same subspace as $\mathbf{w}_p$. The optimization problem for every trajectory is collected and written succinctly in matrix form as

$$\min \|\mathbf{C}\|_1 \quad \text{s.t.} \quad \mathbf{W} = \mathbf{W}\mathbf{C}, diag(\mathbf{C}) = 0. \tag{7}$$

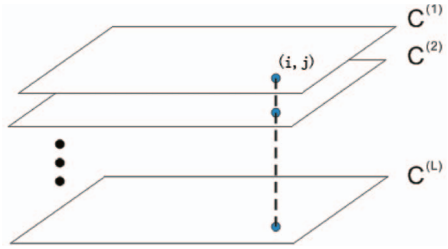

Figure 3. Illustration of the $\ell_{1,1,2}$ norm minimization. The entries (i, j) of $\mathbf{C}^{(l)}$ should be sparse and its support set should be consistent across different $\mathbf{C}^{(l)}$.

where $diag(\mathbf{C})$ are the diagonal entries of the matrix $\mathbf{C}$, and $diag(\mathbf{C}) = 0$ is introduced to avoid the trivial solution.

According to [9], since the optimal solution $\mathbf{C}^*$ to problem (7) measures the pairwise linear correlations among trajectories, it can be naturally used to construct an affinity matrix $\mathbf{A}$ with $\mathbf{A}_{ij} = |\mathbf{C}^*_{ij}| + |\mathbf{C}^*_{ji}|$, after which spectral clustering algorithms can be applied to obtain the desired segmentation into the respective subspaces.

3.1.2 Multiple image pairs

A naive way to extend the SSC algorithm to multi-view case is to compute results from many image pairs individually and design a voting scheme to determine to which group the data points should belong. An alternative way is to accumulate the individual affinity matrices or adopt the multi-view spectral clustering method [36]. However, these methods operate on each image pair separately, and have not exploited the linkage between the multiple image pairs in a more integral manner. Here, we seek to incorporate all image pairs into a unified optimization process.

Assuming we have L image pairs, and since each image pair yields a correspondence matrix $\mathbf{W}^{(l)}$, L corresponding coefficient matrices $\mathbf{C}^{(l)}$ will be constructed by SSC. The key here is to solve for all $\mathbf{C}^{(l)}$ together and require them to share a common sparsity profile. In other words, the non-zero entries of $\mathbf{C}^{(l)}$ should be sparse and those columns corresponding to the same trajectory across the different $\mathbf{C}^{(l)}$ should share the same support set. This amounts to solving a joint sparse optimization problem [21], which can be relaxed into the following mixed norm minimization problem:

$$\begin{aligned} &\min \textstyle\sum_{i=1}^{P} \sum_{j=1}^{P} \sqrt{\sum_{l=1}^{L} (c_{ij}^{(l)})^2} \\ &\text{s.t.} \quad \mathbf{W}^{(l)} = \mathbf{W}^{(l)} \mathbf{C}^{(l)}, \quad diag(\mathbf{C}^{(l)}) = 0, \\ &\qquad l = 1, \ldots, L, \end{aligned} \tag{8}$$

where $c_{ij}^{(l)}$ is the (i, j)-th element of $\mathbf{C}^{(l)}$ for the l-th image pair. Referring to Figure 3, this operation can be visualized as stacking all $\mathbf{C}^{(l)}$ into a tensor $\mathcal{C} \in \mathbb{R}^{P \times P \times L}$, and then minimizing the number of non-zero entries in the aggregate matrix formed by summing all $c_{ij}^{(l)}$ along the third dimension l. In analogy to the $\ell_{1,2}$ norm being the norm that

approximately measures the number of non-zero columns, we can call our norm the $\ell_{1,1,2}$ norm. Denote $\mathcal{C}^*$ as the optimal solution. We similarly construct an affinity matrix $\mathbf{A}$ with its element $\mathbf{A}_{ij} = \sqrt{\sum_{l=1}^{L}(c_{ij}^{*(l)})^2} + \sqrt{\sum_{l=1}^{L}(c_{ji}^{*(l)})^2}$. Then spectral clustering is applied as in the two-frame case.

Notice that the correspondences can be missing in some image pairs, here "missing" means a trajectory is invisible in either one or both of the image pair. In this case, we fill in with a $\mathbf{0}_{9\times 1}$ column vector for the missing data so as to ensure that all $\mathbf{W}^{(l)}$ have the same dimension. More specifically, if a trajectory p is missing in the image pair l, then in the l-th correspondence matrix $\mathbf{W}^{(l)}$, the p-th column $\mathbf{w}_p^{(l)} = \mathbf{0}_{9\times 1}$. Our rationales for filling in with $\mathbf{0}_{9\times 1}$ are twofold: 1) when we want to obtain the sparse coding for the p-th point, the optimal solution for the missing data in the l-th image pair is $\mathbf{0}_{P-1\times 1}$, not incurring any cost in equation (8), nor biasing the solution for other $\mathbf{C}^{(l)}$ in any way. 2) Conversely when we want to recover the sparse coding for other points, e.g. q, the missing data will not be chosen to represent the point q in the l-th image pair since it contributes nothing to the representation of q. This allows us to treat a trajectory with missing data in a uniform manner, without affecting the joint optimization scheme.

3.1.3 Handling ambiguous matches

In real applications, feature trackers often bring in noisy or even heavily corrupted trajectories, especially if we want to seek a denser coverage of features over the entire image. In order to recover the sparse coefficients from the corrupted observations, it is straightforward to consider the following regularized minimization problem:

$$\begin{aligned} &\min \textstyle\sum_{i=1}^{P}\sum_{j=1}^{P}\sqrt{\sum_{l=1}^{L}(c_{ij}^{(l)})^2} + \lambda\sum_{l=1}^{L}\|\mathbf{E}^{(l)}\|_{\ell} \\ &\text{s.t.} \quad \mathbf{E}^{(l)} = \mathbf{W}^{(l)} - \mathbf{W}^{(l)}\mathbf{C}^{(l)}, \quad diag(\mathbf{C}^{(l)}) = 0, \\ &\qquad\qquad l = 1, \ldots, L, \end{aligned} \tag{9}$$

where λ is a weight used to adjust the effect of the two parts and $\|\cdot\|_{\ell}$ indicates a particular choice of regularization strategy. Here we choose $\ell_{1,2}$ norm to model sample-specific corruptions and outliers [19], whose minimization forces $\mathbf{E}^{(l)}$ to be column sparse.

After obtaining an optimal solution $(\mathcal{C}^*, \mathcal{E}^*)$ (where $\mathcal{E}^* \in \mathbb{R}^{D\times P\times L}$ is a tensor stacked from $\mathbf{E}^{*(l)}$), we could detect erroneous matches by looking for those columns with large ℓ_2 norms in any of the $\mathbf{E}^{*(l)}$. If a corrupted match is detected in $\mathbf{E}^{*(l)}$, we will delete it from image pair l but preserve the correct matches of that trajectory in other image pairs unless all matches of that trajectory are corrupted.

3.2. Merging via coefficient analysis

As the number of motion groups is usually not known *a priori* in reality, we have to come to grips with the model selection problem. In view of the difficulty of cluster detection, we propose to first over-segment the data based on the number of zero eigenvalues of the Laplacian matrix of the affinity matrix, and then attempt to merge the clusters later via the following model selection scheme.

Given a data point $\mathbf{q} \in \mathbb{R}^D$ and a group of points $\{\mathbf{p}_i\}_{i=1}^{M}$ stacked as the columns of the matrix $\mathbf{P} \in \mathbb{R}^{D\times M}$ and spanning the subspace $\mathcal{S}$, if we use $\mathbf{P}$ to represent $\mathbf{q}$, *i.e.* $\mathbf{q} = \mathbf{P}\mathbf{c}$, we can obtain a coefficient vector $\mathbf{c} \in \mathbb{R}^M$. According to Theorem 1.3 of [26], the data point $\mathbf{q}$ has a high probability of being an outlier w.r.t. $\mathcal{S}$ if the ℓ_1-norm of the sparsest solution $\mathbf{c}$ is larger than a threshold $\epsilon = \lambda(\frac{M-1}{D})\sqrt{D}$ (λ is a threshold ratio function; for details, see [26]). Based on this theorem, we can determine the relationship between two groups.

Now consider two groups of points obtained from the over-segmentation step, $\mathbf{P} \in \mathbb{R}^{D\times M}$ and $\mathbf{Q} \in \mathbb{R}^{D\times N}$, whose columns $\{\mathbf{p}_i\}_{i=1}^{M}$ and $\{\mathbf{q}_i\}_{i=1}^{N}$ are extracted from subspaces $\mathcal{S}_u$ and $\mathcal{S}_v$ respectively. If we sparsely represent the points in $\mathbf{P}$ using the points in $\mathbf{Q}$:

$$\begin{aligned} &\min \|\mathbf{C}\|_1 \\ &\text{s.t.} \quad \mathbf{P} = \mathbf{Q}\mathbf{C}, \end{aligned} \tag{10}$$

the columns of $\mathbf{C} \in \mathbb{R}^{N\times M}$ are the coefficient vectors corresponding to the data points in $\mathbf{P}$. Based on the aforementioned outlier determination scheme, if $u = v$ and $\mathbf{Q}$ adequately represents $\mathcal{S}_v$, the points in $\mathbf{P}$ should be inliers w.r.t. $\mathbf{Q}$, and thus the ℓ_1-norms of columns $\{\mathbf{c}_i\}_{i=1}^{M}$ in $\mathbf{C}$ are expected to be small. For robustness, we compare the median value of all ℓ_1-norms of $\{\mathbf{c}_i\}_{i=1}^{M}$ against the threshold ϵ to decide if $\mathbf{P}$ should be merged into $\mathbf{Q}$. For notational convenience, we denote the above using a relationship matrix $\mathbf{R}$ with its elements defined as

$$\mathbf{R}_{pq} = median_{i=1}^{M}(\|\mathbf{c}_i\|_1). \tag{11}$$

Similarly, we can obtain $\mathbf{C}' \in \mathbb{R}^{M\times N}$ by representing $\mathbf{Q}$ using $\mathbf{P}$ and compute the relationship $\mathbf{R}_{qp}$. Note that this relationship is oriented, and in general, $\mathbf{R}_{pq} \neq \mathbf{R}_{qp}$.

The above analysis can be extended to the case for the multiple image pairs in a manner analogous to the collaborative clustering algorithm in (8). Assuming L image pairs, we rewrite (10) as

$$\begin{aligned} &\min \textstyle\sum_{i=1}^{N}\sum_{j=1}^{M}\sqrt{\sum_{l=1}^{L}(c_{ij}^{(l)})^2} \\ &\text{s.t.} \quad \mathbf{P}^{(l)} = \mathbf{Q}^{(l)}\mathbf{C}^{(l)}, \quad diag(\mathbf{C}^{(l)}) = 0, \\ &\qquad\qquad l = 1, \ldots, L, \end{aligned} \tag{12}$$

where $\mathbf{P}^{(l)}$ and $\mathbf{Q}^{(l)}$ are the data matrices of the two groups in the l-th image pair, $\mathbf{C}^{(l)}$ is the corresponding coefficient matrix, and $c_{ij}^{(l)}$ is the (i,j)-th element of $\mathbf{C}^{(l)}$. The relationship $\mathbf{R}_{pq}$ (11) is also changed accordingly:

$$\mathbf{R}_{pq} = median_{i=1}^{M}(median_{l=1}^{L}(\|\mathbf{c}_i^{(l)}\|_1)). \tag{13}$$

We iteratively merge two groups according to the aforesaid threshold ϵ until there is no more merging possible. The details of the merging step are summarized in Algorithm 1.

Algorithm 1 ℓ_1-norm based merging

Input: Set of motion groups $\{\mathbf{P}_k\}_{k=1...K}$, ϵ
  $\mathcal{P}_0 \leftarrow$ Current set of groups
  for $k = 1 \rightarrow (K-1)$ **do**
    for each group pair **do**
      Compute relationship matrix $\mathbf{R}$ according to (13).
    end for
    if $\min(\mathbf{R}) < \epsilon$ **then**
      1.$(i, j) = find(\min(\mathbf{R}))$
      2.Merge the groups i and j
      3.$\mathcal{P}_k \leftarrow$ Current set of groups
    else
      return $\mathcal{P}_k$
    end if
  end for
  return $\mathcal{P}_k$

One might question what if some of the groups are too small or degenerate such that they do not adequately represent the underlying subspace $\mathcal{S}$. Clearly, such groups are common occurrences, but it is also true that there invariably exist some other groups whose points fully span the subspace $\mathcal{S}$. In such cases, the former will be judged to belong to and merged into the latter. [1]

4. Experiments

4.1. Results on single image pairs

In this subsection, we evaluate the performance of the two-frame version of our algorithm on the *Hopkins155* database (denoted as TPV in Tables 1) to gauge the effectiveness of our two-frame method. We compute the classification error as the percentage of misclassified points w.r.t. the ground truth and list the average classification errors. We choose the first and the last frames of all sequences as the image pair for the testing, which ensures that all correspondences in the scene have sufficient displacements in the image plane. For the sake of comparison, we assume the number of motion groups is known in this experiment, like what many algorithms did. We also list the classification errors when applying ALC[22], GPCA[31], LSA[34], SSC[9] and LRR[19] to the affine motion subspace for comparison.

It can be seen from Table 1 that TPV yielded average classification errors of less than 5% for the two and three motions, which is only slightly worse off than those of SSC and LRR applied to multiple views assuming affine model.

[1] Even if a motion group consists of say, just two walls, the degenerate case of the over-segmentation yielding two walls cleanly (and thus not mergeable) seldom arises; instead, the points of the two walls are usually segmented non-exactly by our over-segmentation step.

Table 1. Classification errors (%) on *Hopkins155*

Method	ALC	GPCA	LSA	SSC	LRR	TPV
2 motions: 120 sequences						
Mean	2.40	4.59	3.45	0.82	1.33	1.57
3 motions: 35 sequences						
Mean	6.69	28.66	9.73	2.45	2.51	4.98
All: 155 sequences						
Mean	3.36	10.02	4.86	1.18	1.59	2.34

The results indicate that segmentation from two properly chosen views is almost as good as segmentation from the multiple views. What is noteworthy is that the 2-frame T-PV algorithm outperforms the multi-frame GPCA and LSA algorithms on all categories. We believe that this is due to a combination of factors such as the better modeling of perspective effect and the choice of better clustering methods.

4.2. Results on multiple image pairs

We now evaluate the complete algorithm using multiple image pairs without knowing the number of motion groups and with challenges like missing data and perspective effects. The data used in this evaluation comprise 62 video sequences, of which 50 are from *Hopkins155*. Since *Hopkins155* has a very unbalanced number of 2-motion and 3-motion clips (120 and 35 respectively), we retain only the 50 original seed videos (the other 105 2-motion clips are created by splitting off from the 3-motion clips). More importantly, to evaluate the performance under missing data and perspective effects, we added 12 clips with incomplete trajectories, of which 4 are from [25] and the other 8 are captured by us using a handheld camera with a wide angle lens. The newly captured sequences contain about 100 frames each, some of which experience heavy occlusions, posing significant challenge to the matrix completion task, as we shall see later. Of the resultant 62 motion clips, 26 contain two motions, 36 contain three motions, 12 suffer from missing data, and 9 have strong perspective effects (some of these categories are not mutually exclusive). We refer to this combined dataset as the 62-clip dataset.

We denote our complete algorithm as M-TPV for multiple-TPV. We compare the performance of M-TPV to seven state-of-the-art approaches: ALC[22], GPCA[32], LBF[35], LRR[19], MSMC[8], ORK[6] and SSC[9]. For ALC, we use the provided rather simple matrix completion method and test 101 different values from $1e^{-5}$ to $1e^{3}$ for the noisy level as in [22], and then we record the best segmentations with the smallest average error rate. For MSMC, since the default scales (the number of interval frames between an image pair, with the default scales being h_1, h_5 and h_{25}) did not perform well in these sequences, we tried several combinations and report the error rates corresponding to the following scales: h_5, h_{10} and h_{25}. For SSC, since the model selection method based on spectral gap[26] performed poorly in these real data, we choose the

Table 2. Classification results on 62-clip dataset

Method	ALC	GPCA	LBF	LRR	MSMC	ORK	SSC	M-TPV
Classification error (%) - clips with perspective effect: 9 clips								
Mean	16.18(0.35)	43.66(40.83)	20.00(12.14)	16.31(14.83)	19.17(0.58)	22.94(20.24)	25.68(9.68)	**8.20**(0.46)
Classification error (%) - clips with missing data: 12 clips								
Mean	25.38(0.43)	39.64(28.77)	20.17(18.47)	26.03(29.46)	14.64(1.06)	24.11(22.33)	27.41(17.22)	**7.71** (0.91)
Classification error (%) - clips without missing data: 50 clips								
Mean	22.03(18.28)	16.89(16.20)	15.66(1.90)	9.82(5.26)	14.19(2.59)	12.98(4.15)	13.09(2.01)	**7.56**(2.78)
Classification error (%) - all: 62 clips								
Mean	22.67(14.88)	21.29(16.58)	16.53(5.90)	12.98(5.95)	14.27(2.34)	15.13(8.08)	15.86(5.17)	**7.59**(2.37)
Group number estimation - all 62 clips								
# correct	21	33	29	35	25	37	33	**46**

second order difference (SOD) method as in LBF. Note that the SOD method is also used in a similar manner to support SSC in [35]. For those algorithms which do not explicitly handle missing data, such as LBF, LRR, ORK and SSC, we recover the data matrix using Chen's matrix completion approach[5], which in our experience has the best performance among various competing algorithms (such as OptSpace[20], GROUSE[1] and *etc*.). For those algorithms which have a random element in their results, such as ORK and MSMC, we repeat 100 times and record the best results.

Table 2 shows the performance of these methods on the 62-clip dataset. Since the estimated number of motion groups may not be the same as the ground truth number, we exhaustively test all the cluster pairings to obtain the best error rates. Furthermore, to investigate if good model selection results in good segmentation, the error rates obtained by only considering sequences where the number of motions is correctly estimated are shown in the bracket. We also show some qualitative results obtained with the newly captured clips in Figure 4.

The evaluation in Table 2 can be divided into three parts. In the first part, the classification error rates of the 9 clips with strong perspective effects are presented. Our method is the only one with an error rate of less than 10%, which shows the superiority of the proposed approach. Although ALC and MSMC also reported good results when the number of motion groups is correctly estimated, perspective effects have a significant detrimental impact on their model selection steps, resulting in substantially higher error rates of ALC and MSMC. In the second part of Table 2, the impact of missing data is investigated. Our approach again outperformed the other methods with a less than 10% error rate. GPCA broke down mainly due to the instability of the Power Factorization method used for filling in missing data. Those methods based on the matrix completion of [5] for filling in, such as LBF, ORK and SSC, performed well in some sequences, but the overall deleterious impact is evident, attesting to the difficulty faced by a general-purpose matrix completion algorithm in dealing with the structured pattern of the missing data. Among these methods, it is also remarkable that the so-far top-performing LRR failed in the model selection of 11 sequences, which implies that the model selection step in LRR is very sensitive to how the spectral values have been changed in the recovered matrix. Of the only sequence whose motion number is correctly estimated (the "Van" clip, last row of Figure 4), LRR has a very poor classification error rate. MSMC failed in those sequences with complicated objects and backgrounds due to its simple motion model based on homography. Even if this method uses a higher-order motion model, the significant increase in model complexity will pose a lot of difficulties for the sampling procedure, rendering its performance very much suspect. The last comparison is based on the 50 seed videos from the *Hopkins155*'s dataset. These clips are relatively easy, because they have complete trajectories. The average classification error of our method on all 50 clips is 7.56%, while that considering only cases having correct motion number estimation is 2.78%. The more meaningful figure of 7.56% is clearly the best compared to other state-of-the-art motion segmentation algorithms. These figures also demonstrate that model selection remains a recalcitrant problem, and to achieve real progress in motion segmentation, we must meet this challenge heads-on.

The last two rows of metrics in Table 2 measure the overall performance, from which it can be seen that our method outperformed the rest in all significant aspects. It has 46 correct motion number estimation out of 62 clips (next best is 37), and the average classification error of all clips is 7.59% (next best is 12.98%). These overall performances demonstrate that our method is capable of handling the various real challenges in the motion segmentation problem.

5. Conclusions

We solve the 3D motion segmentation problem of multiple frames rooted in the epipolar geometry of two perspective views via a collaborative clustering algorithm. This approach highly integrates multiple frame information with a mixed norm optimization, which is able to avoid the disadvantages of multi-frame methods and enjoy the rich information provided by multiple frames. We also propose a

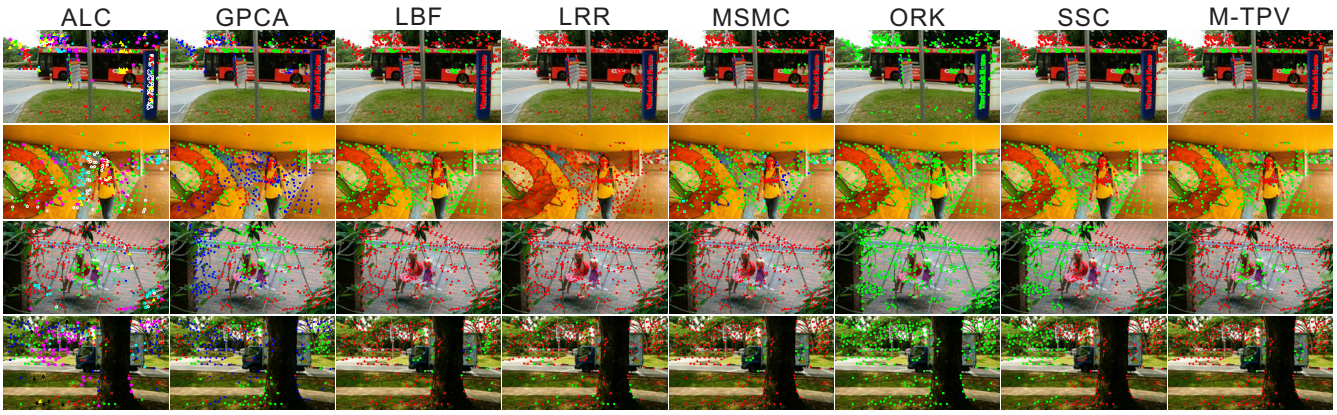

Figure 4. Qualitative results of the real data with missing entries. The segmentation results of the 50-th frames of the sequences are presented. From top to bottom are the "Bus", "Girl", "Swing" and "Van" clips.

method to evaluate the relationship of two groups based on a similar optimization scheme. Leveraging on this, we first over-segment the motion groups, and then merge them according to the relationships. The experiments on the *Hopkins155* database and the new sequences showed that the proposed algorithm outperforms the state-of-the-art methods in meeting the various challenges.

Acknowledgements. This work was partially supported by the Singapore PSF grant 1321202075 and the grant from the National University of Singapore (Suzhou) Research Institute (R-2012-N-002).

References

[1] L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. Technical report, arXiv:1006.4046, 2011. 7

[2] T. Boult and L. Brown. Factorization-based segmentation of motions. In *Proc. of the IEEE Workshop on Motion Understanding*, 1991. 1, 2

[3] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010. 3

[4] E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Inform. Theory*, 56(5):2053–2080, 2009. 2

[5] P. Chen. Optimization algorithms on subspaces: Revisiting missing data problem in low-rank matrix. *IJCV*, 80(1):125–142, 2008. 2, 7

[6] T.-J. Chin, H. Wang, and D. Suter. Robust fitting of multiple structures: The statistical learning approach. In *ICCV*, 2009. 3, 6

[7] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *IJCV*, 29(3):159–179, 1998. 1, 2

[8] R. Dragon, B. Rosenhahn, and J. Ostermann. Multi-scale clustering of frame-to-frame correspondences for motion segmentation. In *ECCV*, 2012. 3, 6

[9] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *CVPR*, 2009. 1, 2, 4, 6

[10] C. W. Gear. Multibody grouping from motion images. *IJCV*, 29(2):133–150, 1998. 1, 2

[11] A. Gruber and Y. Weiss. Multibody factorization with uncertainty and missing data using the em algorithm. In *CVPR*, 2004. 1, 2

[12] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 3

[13] Y. Jian and C. Chen. Two-view motion segmentation with model selection and outlier removal by ransac-enhanced dirichlet process mixture models. *IJCV*, 88(3):489–2501, 2010. 1, 3

[14] K. Kanatani. Motion segmentation by subspace separation and model selection. In *ICCV*, 2001. 1, 2

[15] J. Lezama, K. Alahari, J. Sivic, and I. Laptev. Track to the future: Spatio-temporal video segmentation with long-range motions cues. In *CVPR*, 2011. 3

[16] H. Li. Two-view motion segmentation from linear programming relaxation. In *CVPR*, 2007. 1, 3

[17] T. Li, V. Kallem, D. Singaraju, and R. Vidal. Projective factorization of multiple rigid-body motions. In *CVPR*, 2007. 3

[18] Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical report, Technical report, UILU-ENG-09-2215, 2009. 2

[19] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE TPAMI*, (99):1, 2012. 1, 2, 3, 4, 5, 6

[20] A. Montanari and S. Oh. Matrix completion from a few entries. *IEEE Trans. Inform. Theory*, 56(6):2988–2998, 2010. 7

[21] A. Rakotomamonjy. Surveying and comparing simultaneous sparse approximation (or group-lasso) algorithms. *Signal Processing*, 91(7):1505–1526, 2011. 4

[22] S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE TPAMI*, 32(10):1832–1845, 2010. 1, 2, 6

[23] S. Rao, A. Yang, S. Sastry, and Y. Ma. Robust algebraic segmentation of mixed rigid-body and planar motions from two views. *IJCV*, 88(3):425–446, 2010. 1, 3

[24] K. Schindler and D. Suter. Two-view multibody structure-and-motion with outliers through model selection. *IEEE TPAMI*, 28(6):983–995, 2006. 1, 3

[25] K. Schindler, J. U, and H. Wang. Perspective n-view multibody structure-and-motion through model selection. In *ECCV*, 2006. 3, 6

[26] M. Soltanolkotabi and E. J. Candès. A geometric analysis of subspace clustering with outliers. *Annals of Statistics*, 40(4):2195–2238, 2011. 2, 3, 5, 6

[27] Y. Sugaya and K. Kanatani. Geometric structure of degeneracy for multi-body motion segmentation. In *Workshop on Statistical Methods in Video Processing*, 2004. 1, 2

[28] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography. *IJCV*, 9(2):137–154, 1992. 1, 2

[29] P. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *IJCV*, 50(1):35–61, 2002. 1, 3

[30] R. Tron and R. Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *CVPR*, 2007. 1

[31] R. Vidal and R. Hartley. Motion segmentation with missing data by power factorization and generalized pca. In *CVPR*, 2004. 1, 2, 6

[32] R. Vidal, Y. Ma, S. Soatto, and S. Sastry. Two-view multibody structure from motion. *IJCV*, 68(1):7–25, 2006. 1, 3, 6

[33] L. Wolf and A. Shashua. Two-body segmentation from two perspective views. In *CVPR*, 2001. 1, 3

[34] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV*, 2006. 1, 2, 6

[35] T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Hybrid linear modeling via local best-fit flats. *IJCV*, 100(3):217–240, 2012. 6, 7

[36] D. Zhou and C. Burges. Spectral clustering and transductive learning with multiple views. In *ICML*, 2007. 4