# A NEW TRAJECTORY CLUSTERING ALGORITHM USING TEMPORAL SMOOTHNESS FOR MOTION SEGMENTATION

*F. Shi*[1], *Z. Zhou*[1,*], *J. Xiao*[2], *W. Wu*[1]

[1]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, China
[2]Ningbo Industrial Technology Research Institute, CAS, China

## ABSTRACT

In this paper, a new trajectory clustering algorithm for motion segmentation is proposed. Our key contribution is to use temporal smoothness constraint to facilitate segmentation of incomplete trajectories, which leads to high robustness to missing data. We further show that most motions in foreground of a scene can be approximately represented by a set of translational motion models. Based on this assumption, a new clustering strategy is proposed to separate foreground objects from background. Finally, a series of experiments show that our approach is more effective and outperforms several state-of-the-art methods.

***Index Terms***— Motion Segmentation, Trajectory Clustering, Temporal Smoothness

## 1. INTRODUCTION

Segmenting a video into multiple temporally consistent clusters according to their motions is referred to as motion segmentation. The importance and the variety of the possible applications make the problem be an active topic in computer vision. In this paper, we focus on sparse methods in motion segmentation, $i.e.$ trajectory clustering algorithms. Please see Figure 1 as an illustration of trajectory clustering algorithm.

In this field, the most important class of algorithms is multi-body factorization methods [1, 2, 3, 4, 5], and their underlying idea is using motion subspaces constraints, where the trajectories of the same motion can span a low-dimensional linear subspace and different motions may distribute in different subspaces. Based on this fact, segmenting a video containing various types of motion ($e.g.$ independent, articulated, rigid, non-rigid or any combination of them) can be cast as a subspace separation problem, and thus can be solved in a unified way. However, this kind of algorithm has an inherent drawback, which requires an assumption that each motion should have a sufficiently large set of complete trajectories. If input data is highly fragmented, which are common phenomena in real world tracking, the performance of multi-body factorization methods will deteriorate drastically. In recent years, a few clustering methods [6, 7, 8, 9] which do not require any completion of trajectories are proposed for motion segmentation. These methods measure similarities between trajectories based on a motion model, and then employ a common clustering technique, such as spectral clustering, to segment trajectories. Compared to multi-body factorization methods, these motion model-based methods have significant advantage in handling incomplete trajectories, but require more accurate motion models. It will often lead to poor performance when they were applied to segment a video containing motions that deviate from underlying motion model.

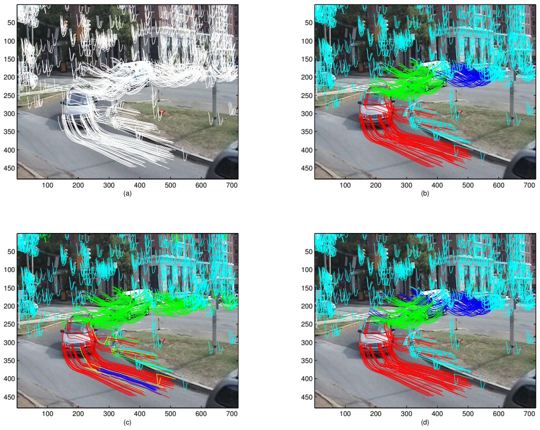


**Fig. 1**. Trajectory clustering algorithm applied on the video 'carsTurning' of the Hopkins 155+16 dataset. (a) Input trajectories, (b) ground-truth segmentation, different colors indicate different motions in the video, (c) clustering result of [4], (d) clustering result of our algorithm.

We propose a new trajectory clustering algorithm for motion segmentation that combines advantages of the two kinds of algorithms described above. Our method is highly robust to missing data, and possesses ability to segment video sequences that include various types of motion. The first step of our method is to decompose the input trajectories into a set of Discrete Cosine Transform (DCT) bases and corresponding coefficients with a non-linear optimization scheme. By this way, we can exploit temporal smoothness of trajecto-

[*] Corresponding author. E-mail: zz@vrlab.buaa.edu.cn.

ries to compactly approximate the bases of trajectories with predefined vectors. This results in a significant reduction in unknowns, and increases robustness in handling incomplete trajectories. Another benefit of using DCT basis is that it makes the coefficients of trajectories to be an effective way to measure trajectory similarities. Therefore, we next perform cluster analysis on the coefficients of trajectories. We further observe that trajectories belonging to the same foreground object in a scene can be approximately described by a translational motion model due to spatial proximity. Then based on this assumption, a new clustering strategy is presented, where we first separate foreground trajectories from background and then divide the foreground trajectories into different partitions using a translational model-based clustering method. Finally, we evaluate our approach on the Hopkins 155+16 dataset [10], and obtain more accurate segmentation results than existing motion segmentation algorithms.

## 2. PROPOSED ALGORITHM

In this paper, we suppose the trajectories of feature points have been obtained from some existing trackers such as KLT tracker [11] on the video.

### 2.1. Matrix Factorization of Trajectory Data

Given trajectories of $P$ tracked feature points in a video with $F$ frames, we use $T(p) = (x_1^p, ..., x_F^p, y_1^p, ..., y_F^p)^T$ to denote the $p^{th}$ trajectory, where $x_f^p$ and $y_f^p$ are the $X$ and $Y$ coordinates of the $p^{th}$ point at frame $f$. Our goal is to partition the $P$ trajectories into different groups according to their corresponding motions. We first form a measurement matrix $W \in R^{F\times 2P}$ by arranging the $X$ and $Y$ coordinates of all trajectories vertically:

$$W = \begin{pmatrix} x_1^1 & y_1^1 & \cdots & x_1^P & y_1^P \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_F^1 & y_F^1 & \cdots & x_F^P & y_F^P \end{pmatrix} \tag{1}$$

Assuming rank($W$) = $r$, $W$ can then be decomposed as: $W = BC$, where the columns of $B \in R^{F\times r}$ are the bases of the column space of $W$, and the $2p^{th} - 1$ and $2p^{th}$ columns of $C \in R^{r\times 2P}$ are the corresponding coefficients of the $X$ and $Y$ coordinates of $T(p)$. As discussed below, when appropriate bases are chosen, the coefficients of $T(p)$ can provide a good measure of trajectory similarity. Thus, we intend to perform cluster analysis on the trajectory coefficients, and then to label the trajectories accordingly.

**Algorithm 1** Alternated Least-Squares algorithm to factorize $W$

```
Input $X = [I_r, 0]^T$.
Repeat
    Compute $C$ with Eq. (4).
    Compute $B$ with Eq. (5), then $X = (\Omega_d)^T B$.
    Orthogonalize the columns of $X$.
Until convergence.
```

Generally speaking, due to occlusions and tracker limitations, there usually have some incomplete trajectories in the input to our algorithm. It may result in a particular challenge in the pre-step of clustering trajectory coefficients: factorizing $W$ with missing data. To solve this problem, we propose to use temporal smoothness constraint of trajectories to exploit inherent property of the natural deforming objects, which has been successively used in algorithms that reconstruct scene from tracked feature points [12, 13]. In our setup, the temporal smoothness of $T(p), p = 1, ..., P$ suggests that the values in each column of $W$ vary smoothly over time, thus can be considered as samples of a smooth signal. This means that there exist a number of predefined bases which can approximate the columns of $W$ compactly, and therefore results in a significant reduction in unknowns and corresponding accuracy of estimation of trajectory coefficients. Here, considering the effectiveness of the DCT basis in representing motion trajectories [12, 13], we use a linear combination of $d$ ($r < d \ll F$) DCT vectors to approximate the column bases of $W$. Then $W$ can be factorized as:

$$W = BC = \Omega_d XC = \begin{pmatrix} \theta^1 & \cdots & \theta^d \end{pmatrix} XC \tag{2}$$

where $\theta^j$ denotes the $j^{th}$ DCT basis, and its $i^{th}$ component is denoted by $\theta_i^j$ as:

$$\theta_i^j = \frac{\sigma_j}{\sqrt{F}} \cos(\frac{\pi(2i-1)(j-1)}{2F}), \sigma_1 = 1, \sigma_j = \sqrt{2}, j \geq 2 \tag{3}$$

Let $w^p$ and $w_f$ denote the $p^{th}$ column and $f^{th}$ row of $W$, and $c^p$ and $b_f$ denote the $p^{th}$ column and $f^{th}$ row of $C$ and $B$. In the case of missing data, let $\tilde{w}^p$ and $\tilde{w}_f$ denote the observed entries in $w^p$ and $w_f$. We then use an Alternated Least-Squares (ALS) algorithm described in Algorithm 1 to factorizes $W$.

The Eq. (4) and (5) in Algorithm 1 are of the form:

$$vec(C) = (\Psi^T \Psi)^{-1} \Psi^T vec(\tilde{W}) \tag{4}$$

where $\Psi$ is a block diagonal matrix which is formed by $\Pi^p \Omega_d X, p = 1, ..., 2P$, $\Pi^p$ is defined as a row-amputated identity matrix such that $\Pi^p \Omega_d X$ has the rows in $\Omega_d X$ that correspond to the rows of entries in $\tilde{w}^p$, $vec(\tilde{W}) = ((\tilde{w}^1)^T, ..., (\tilde{w}^{2P})^T)^T$, $vec(C) = ((c^1)^T, ..., (c^{2P})^T)^T$.

$$vec(B^T) = (\Phi^T \Phi)^{-1} \Phi^T vec(\tilde{W}^T) \tag{5}$$

where $\Phi$ is a block diagonal matrix which is formed by $(\Pi_f)^T C^T, f = 1, ..., F$, $\Pi_f$ is defined as a column-amputated identity matrix such that $C\Pi_f$ has the columns in $C$ that correspond to the columns of entries in $\tilde{w}_f$, $vec(\tilde{W}^T) = (\tilde{w}_1, ..., \tilde{w}_F)^T$, $vec(B^T) = (b_1, ..., b_F)^T$.

After $X$ and $C$ are computed, we denote $XC$ by $S = ((s_1)^T, ..., (s_d)^T)^T$ and, from Eq. (2), obtain:

$$((s_1)^T, ..., (s_d)^T)^T = ((\theta^1)^T, ..., (\theta^d)^T)^T W \tag{6}$$

Then, the first row of $S$ equals to

$$s_1 = (\theta^1)^T W = \sqrt{F}(\overline{w^1}, \overline{w^2}, ..., \overline{w^{2P-1}}, \overline{w^{2P}}) \qquad (7)$$

where $(\overline{w^{2P-1}}, \overline{w^{2P}}) = (\sum_{f=1}^{F} x_f^p, \sum_{f=1}^{F} y_f^p)/F$ is the centroid of $T(p)$. Thus, $s_1$ can represent the average spatial location of $T(p), p = 1, ..., P$. Subsequently, combining Eq. (7) with (2), we have:

$$\sum_{j=2}^{d} \theta^j s_j = W - \theta^1 s_1 = (w^1 - e\overline{w^1}, ..., w^{2P} - e\overline{w^{2P}}) \quad (8)$$

where $e = (1, ..., 1)^T \in R^F$. Note that, the $2P^{th} - 1$ and $2P^{th}$ columns of $\sum_{j=2}^{d} \theta^j s_j$ can capture the variation in $T(p)$ relative to its centroid. Thus, $\sum_{j=2}^{d} \theta^j s_j$, and hence $(s_2^T, ..., s_d^T)^T$, can represent the $P$ trajectories' local variation around the average spatial location.

Next, utilizing SVD, we decompose $S$ into $U \in R^{d \times r}$, $D \in R^{r \times r}$ and $V^T \in R^{r \times 2P}$, and specify the matrix $B$ and $C$ as $\Omega_d UD$ and $V^T$ respectively. Note that, the coefficients matrix $C$ is now given by $(D^{-1}U^T)S$. It follows that the coefficients of $T(p)$, $i.e.$ $(c^{2p-1}, c^{2p})$, are just the weighted sum of its average spatial location and its local variation around that location. As a consequence, $C(p) = ((c^{2p-1})^T, (c^{2p})^T)^T$ can be seen as the integration of spatial location and motion pattern of $T(p)$, and thus can be used to distinguish trajectories belonging to different motions. So, in the next subsection, we are going to perform clustering on $C(p), p = 1, ..., P$, and then to label the trajectories accordingly.

### 2.2. Divisive Clustering of Trajectory Coefficients

Given vectors that can measure trajectory similarities, a common way to cluster them is to compute pairwise distances between all vectors firstly, and then to analyze the pairwise distances with spectral clustering or agglomerative clustering. In fact, this pairwise analysis can only compare the similarity of trajectories on the basis of translational motion models, and Brox *et al.* [7] indicated that, relying on the fact that translational models are a good approximation for spatially close points, the pairwise analysis can also be used to segment videos that contain non-translational motions. However, we find that feature points of background of a scene are usually distributed over a wide area, thus their non-translational motions cannot be approximately represented by a translational model. As a result, poor performance of the pairwise analysis based clustering algorithms will often be triggered when they were applied to video captured from a freely moving camera. To circumvent this problem, we have developed the following divisive clustering algorithm, which first separates foreground from background based on motion subspaces constrains then segments foreground using pairwise analysis, to segment $C(p), p = 1, ..., P$:

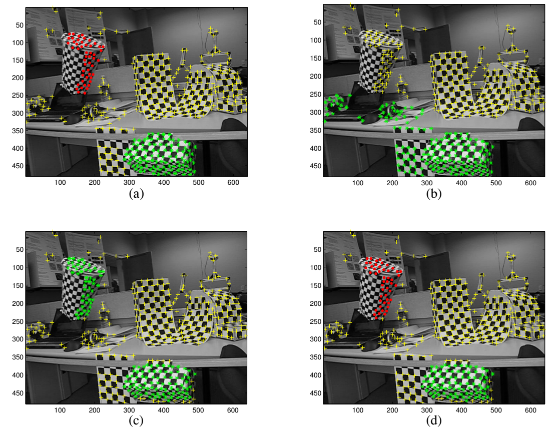


**Fig. 2**. The segmentation results of our method on the video '1R2RCR' of the Hopkins 155 dataset. (a) ground-truth segmentation, object with red dots and with green stars denote two clusters in foreground while with yellow pluses denotes the background cluster, (b) result of step 2 of our clustering algorithm, (c) result of step 3 of our clustering algorithm, (d) final result of our clustering algorithm. For better visualization, we only render the feature points in one frame.

1. Compute affinity matrix $A$ for $C(p), p = 1, ..., P$:

$$A(i, j) = exp(-\|C(i) - C(j)\|_2) \qquad (9)$$

2. Apply spectral clustering to $A$ to segment all trajectories into 2 clusters, and choose the one with lower dimension as background[1].

3. Iterate the following two steps until convergence:

   (a) Compute the bases of background subspace by performing SVD on the matrix formed by $C(p), p \in background : N = (\mu_1, \mu_2, \mu_3, \mu_4)$.

   (b) Compute projection error of trajectories to background subspace: $\epsilon(p) = \|(I_{2r} - N(N)^+)C(p)\|_2$, then apply K-means to $\epsilon(p)$ to repartition all trajectories into foreground and background[2].

4. Compute affinity matrix for $C(p), p \in foreground$ with Eq. (9), and apply recursive 2-way Ncut [14] to the affinity matrix to generate clusters in foreground[3].

Figure 2 illustrates a series of segmentation results of our method on the video sequence '1R2RCR' from the Hopkins 155 dataset.

[1] The dimensions of the clusters are computed by SVD.

[2] $N^+$ denoting the Moore-Penrose pseudo-inverse of the matrix $N$.

[3] The number of clusters in foreground is assumed to be known a priori.

| Method | GPCA | ALCsp | SSC-N | Our Method |
|---|---|---|---|---|
| Mean | 10.34% | 3.56% | 1.24% | **0.86%** |
| Median | 2.54% | 0.50% | 0.00% | **0.00%** |

**Table 1**. Classification errors for the Hopkins 155 dataset. ALCsp and SSC-N represent ALC with sparsity-preserving projection and SSC with Normal random projection.

| Method | GPCA | ALCsp | SSC-N | Our Method |
|---|---|---|---|---|
| 12 sequences with missing data, missing: 4%-35% | | | | |
| Mean | 14.94% | 1.28% | **0.13%** | **0.16%** |
| Median | 9.32% | 1.07% | **0.00%** | **0.08%** |
| 4 sequences with missing data, missing: 15%-60% | | | | |
| Mean | 46.53% | 14.04% | 18.13% | **3.16%** |
| Median | 42.01% | 12.62% | 19.84% | **2.49%** |

**Table 2**. Classification errors for 16 additional Hopkins video sequences.

## 3. EXPERIMENTS

In this section, we evaluate our method on both the Hopkins 155 dataset [10] and 16 video sequences that are complement of the standard Hopkins dataset[4] by comparing with state-of-the-art motion segmentation algorithms. In all experiments, we set $d = max(min(0.1 \times F, 15), 5)$, and choose $r$ from 2 to 10 to provide the best results.

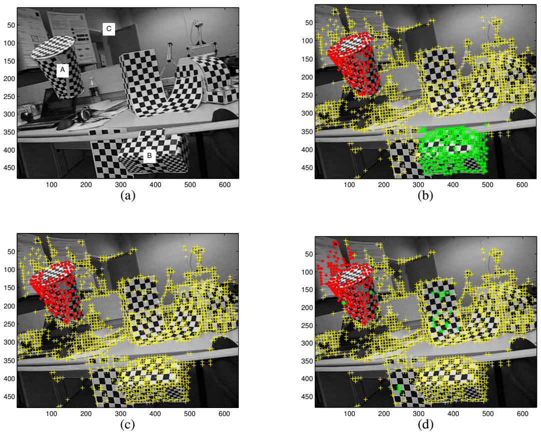
### 3.1. Quantitative Evaluation

We first evaluate our method on the Hopkins 155 dataset by comparing with GPCA [3], ALC [4] and SSC [5]. The dataset contains examples of independent, articulated, rigid, and non-rigid motions, and its video sequences don't contain any missing entries. The classification errors (ration of misclassified trajectories to total trajectories) of the four algorithms are shown in Table 1. We then compare our method with GPCA, ALC and SSC on 16 additional Hopkins video sequences that contain missing data, and report the classification errors in Table 2.

As Table 1 and 2 show, our method achieves the best performance among the four algorithms, and when percents of missing entries in video sequences are increasing, the superiority of our method becomes more significant.

### 3.2. Qualitative Evaluation

The above experiments show the advantage of our method: highly robust to missing data and works well on various types of motion. To further demonstrate this point, we make a feature point sequence by running the tracker in [15] on the video '1R2RCR' of the Hopkins 155 dataset. This video sequence contains not only a non-translational background motion, but also significant missing data. We then perform our method, Brox&Malik [7] and ALC on it, and results are illustrated in Figure 3. It can be seen that only our method gives correct segmentation result. The failure of ALC is primarily attributed to its limited ability to handle incomplete trajectories, and of Brox&Malik is from the inability of its underlying motion model to describe the background motion.



(a) (b) (c) (d)

**Fig. 3**. The clustering results of our method, Brox&Malik and ALC-miss on the video '1R2RCR' of the Hopkins 155 dataset. (a) frame 16 of '1R2RCR', object overlaid by letter 'A' and 'B' denote two clusters in foreground while by letter 'C' denotes the background cluster, (b) result of our method, (c) result of Brox&Malik, (d) result of ALC-miss. For better visualization, we only render the feature points in one frame.

## 4. CONCLUSION

This paper proposes a new trajectory clustering algorithm for motion segmentation. Compared to existing methods, the innovations of our work include two parts. First, we use temporal smoothness of trajectories to handle incomplete trajectories segmentation. Second, we employ a novel clustering strategy that first separates foreground from background and then partition foreground into different clusters. Experiments show the advantage of our method in terms of robustness to missing data and effective range. Especially, when applied to a video sequence containing both significant occlusions and complex motions, other state-of-the-art motion segmentation algorithms may fail while our method gives expected results.

## 5. ACKNOWLEDGEMENTS

This work is supported by the National 863 Program of China under Grant No.2012AA011803, the Natural Science Foundation of China under Grant No.61170188 and the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant No.20121102130004.

[4] Available from http://www.vision.jhu.edu/data/hopkins155/

## 6. REFERENCES

[1] J. S. Costeira and T. Kanade, “A multibody factorization method for independently moving objects,” *International Journal of Computer Vision*, vol. 29(3), pp. 159–179, September 1998.

[2] J. Yan and M. Pollefeys, “A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate,” in *Proc. ECCV*, 2006, vol. 3954, pp. 94–106.

[3] R. Vidal, R. Tron, and R. Hartley, “Multiframe motion segmentation with missing data using powerfactorization and gpca,” *International Journal of Computer Vision*, vol. 79(1), pp. 85–105, 2008.

[4] S. R. Rao, R. Tron, R. Vidal, and Y. Ma, “Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories,” in *Proc. CVPR*. IEEE, 2008, pp. 1–8.

[5] E. Elhamifar and R. Vidal, “Sparse subspace clustering,” in *Proc. CVPR*. IEEE, 2009, pp. 2790–2797.

[6] M. Fradet, P. Robert, and P. Perez, “Clustering point trajectories with various life-spans,” in *Proc. CVMP*. IEEE, 2009, pp. 7–14.

[7] T. Brox and J. Malik, “Object segmentation by long term analysis of point trajectories,” in *Proc. ECCV*, 2010, vol. 6315, pp. 282–295.

[8] P. Ochs and T. Brox, “Higher order motion models and spectral clustering,” in *Proc. CVPR*. IEEE, 2012, pp. 614–621.

[9] Q. Mo and B. A. Draper, “Semi-nonnegative matrix factorization for motion segmentation with missing data,” in *Proc. ECCV*, 2012, vol. 7578, pp. 402–415.

[10] R. Tron and R. Vidal, “A benchmark for the comparison of 3d motion segmentation algorithms,” in *Proc. CVPR*. IEEE, 2007, pp. 1–8.

[11] J. Shi and C. Tomasi, “Good features to track,” in *Proc. CVPR*. IEEE, 1994, pp. 593–600.

[12] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, “Trajectory space: a dual representation for nonrigid structure from motion,” *IEEE Trans. on PAMI*, vol. 33(7), pp. 1442–1456, July 2011.

[13] P. F. U. Gotardo and A. M. Martinez, “Computing smooth time-trajectories for camera and deformable shape in structure from motion with occlusion,” *IEEE Trans. on PAMI*, vol. 33(10), pp. 2051–2065, October 2011.

[14] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. on PAMI*, vol. 22(8), pp. 888–905, August 2000.

[15] N. Sundaram, T. Brox, and K. Keutzer, “Dense point trajectories by gpu-accelerated large displacement optical flow,” in *Proc. ECCV*, 2010, vol. 6311, pp. 438–451.