

Research Article

Multibody Nonrigid Structure from Motion Segmentation Based on Sparse Subspace Clustering

Wenqing Huang ¹, Qingfeng Hu ¹, Yaming Wang ², and Mingfeng Jiang ¹

¹Computer Vision and Pattern Recognition Laboratory, Zhejiang Sci-Tech University, Xiasha Campus, Hangzhou, China

²Lishui University, Lishui, China

Correspondence should be addressed to Yaming Wang; yamingwang2000@163.com

Received 29 October 2020; Revised 23 February 2021; Accepted 11 March 2021; Published 30 March 2021

Academic Editor: Marco Rocchetti

Copyright © 2021 Wenqing Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Sparse subspace clustering (SSC) is one of the latest methods of dividing data points into subspace joints, which has a strong theoretical guarantee. However, affine matrix learning is not very effective for segmenting multibody nonrigid structure from motion. To improve the segmentation performance and efficiency of the SSC algorithm in segmenting multiple nonrigid motions, we propose an algorithm that deploys the hierarchical clustering to discover the inner connection of data and represents the entire sequence using some of trajectories (in this paper, we refer to these trajectories as the set of anchor trajectories). Only the corresponding positions of the anchor trajectories have nonzero weights. Furthermore, in order to improve the affinity coefficient and strong connection between trajectories in the same subspace, we optimise the weight matrix by integrating the multilayer graphs and good neighbors. The experiments prove that our methods are effective.

1. Introduction

Nonrigid structure from motion (NRSFM) is a hot topic in the field of computer vision. It is aimed at recovering both camera motion and nonrigid structure from two-dimensional images of a monocular camera. At present, there are many methods for solving this problem, which can achieve satisfied results [1–3]. In 2020, Jensen et al. introduce a benchmark and evaluate 18 different methods in sparse NRSfM [4]. However, most of the methods still assume that there is only one nonrigid structure in the scene. Unfortunately, in the real world, scenes tend to be much more complex; for instance, several activities, such as playing a basketball game and walking, could be performed on the same scene simultaneously. This is an example of scenes with multibody NRSFM. Therefore, the assumption of a single object is not practical. Therefore, the study of NRSFM reconstruction cannot be limited to the case of a single object. However, as it is very difficult to reconstruct the NRSFM with a single object, it is even more difficult to reconstruct a multibody NRSFM.

For this problem, we can easily refer to a scene of multiple rigid objects. In the case of multiple rigid objects, the problem can be divided into two independent steps: first, each object is divided from 2D coordinates [5, 6], and then, the latest reconstruction technology is used for reconstruction [7, 8]. Alternatively, we can regard multiple objects in the same scene as a whole, reconstruct the scene uniformly, and then divide the 3D coordinates. This idea can also be applied to multiple nonrigid motion scenes. However, this idea has a high segmentation accuracy requirement; furthermore, because of nonrigidity and overlapping, it is easy to obtain bad segmentation results.

Therefore, in order to improve the segmentation effect, in this study, we propose a multiple nonrigid motion segmentation algorithm based on sparse subspace clustering. Through subspace clustering, we can learn the affine matrix of the natural coding subspace and then obtain the number of deformable objects, different activities, and membership degree of each sample by using the idea of spectral clustering [9, 10]. To improve the segmentation effect of SSC on multibody

NRSFM, we use random hierarchical clustering to select a group of k trajectories as the set of anchor trajectories to represent the entire sequence. At the same time, to reduce the influence of the input parameters on the algorithm and make the algorithm more extensible, we used a method for estimating the number of anchor trajectories, k . Finally, to improve the robustness of the algorithm, we optimize the weight matrix obtained using multilayer graphs [11] and the concept of the good neighbors [12] to improve the result.

2. Related Work

In this section, we will analyze the research status on multibody NRSFM and SSC.

2.1. Multibody Nonrigid Structure from Motion. The multibody nonrigid structure from motion (multibody NRSFM) is the inevitable extension direction of the research on nonrigid motion structure. At present, there are two main methods for analyzing a multibody nonrigid scene. The first is to separate the tasks of segmentation and reconstruction and tackle them separately. This approach can be regarded as the extension of the approach to multiple rigid scenes to multiple nonrigid scenes. In the second category, the clustering algorithm is added to the constraints of the model using the spatiotemporal union method, and the segmentation and reconstruction are processed simultaneously.

Russell et al. [13] proposed a method of simultaneously segmenting a complex dynamic scene containing a mixture of multiple objects into constituent objects and reconstructing a 3D model of the scene by formulating the problem as hierarchical graph-cut-based segmentation, where the whole scene is decomposed into background and foreground objects, with the complex motion of non-rigid or articulated objects modeled as a set of overlapping rigid parts. In 2014, in addition to performing temporal clustering, Zhu et al. [14] regarded modeling motion as a union of temporal subspaces. Although their work still focuses on a single object, it provides ideas for extending the research to multibody NRSFM. In 2016 and 2017, Kumar et al. [15, 16] proposed using the SSC algorithm to establish a spatiotemporal union model and segmented and reconstructed the multibody NRSFM simultaneously. In two nonrigid bodies on the same scene, it achieved good results. Now, with the development of artificial intelligence (AI) [17, 18] and deep learnings, more and more fields begin to contact deep learnings to explore the comparison between the neural network and early traditional algorithms. In reference [19], the neural network is used to deal with NRSfM problems, which proves that the neural network can also deal with NRSfM problems well.

2.2. Sparse Subspace Clustering. On multibody NRSFM scenes, each object is distributed in different subspaces. As time goes on, different feature points move to different trajectories. Although a principal components analysis (PCA) [20, 21] can find the low-dimensional structure in high-dimensional data, it does not take into account the situation where the dataset contains multiple structures. Therefore, for multiple nonrigid segmentations, the best approach is to

select multiple subspaces instead of a single subspace. Therefore, subspace clustering [22] is a good way to solve this problem. The best solution to the problem of subspace clustering is based on a spectrum. First, calculate the similarity between all the feature points. Then, cluster the affine matrix by spectrum. Finally, we can achieve the goal of separating a specified number of low-dimensional subspaces in the data space. With the development of sparse representation, an interesting concept, the attribute of self-expression, has attracted significant attention in the spectrum-based methods:

$$x_i = Xc_i, \quad (1)$$

where $X = \{x_1, x_2, x_3, \dots, x_i, \dots, x_N\} \in \mathbb{R}^{d \times N}$, represents the input data, and $c_i \in \mathbb{R}^{N \times 1}$ represents the weight vector corresponding to x_i . Equation (1) can be interpreted as follows. Each data point can be expressed as a sparse linear combination of other data points in the same subspace. The main representative methods of this part are sparse subspace clustering, low-rank representative (LRR) [23–25], and least square regression (LSR) [26]. In the independent and nonintersecting subspaces [27–29], the SSC has a strong theoretical guarantee in the case of noise and nonnoise. In the presence of noise and disjoint subspace, the LRR cannot be effectively modeled. Compared to the SSC and LRR, the LSR has the least computational cost, when the subspace is independent.

However, despite the theoretical guarantee and empirical success of the SSC, it is necessary to solve Equation (1) that is equal to the total number of data points for each data point. Therefore, when the data scale is large, the cost of solving this least absolute shrinkage and selection operator (LASSO) problem will be high. To solve the problem of the SSC's poor performance on large-scale datasets, the scalable sparse subspace clustering (SSSC) is proposed in [30]. This method reduces the computational cost by randomly selecting a part of datasets as a group and can be successfully extended to large datasets. However, the algorithm is not stable, because it uses a random way to select support points. In reference [31], the author opted to not solve Equation (1) according to the idea of a greedy algorithm, using orthogonal matching pursuit (OMP) [32] instead to obtain the sparse representation of each data point. Although the OMP algorithm does boost the effectiveness of the SSC algorithm on large-scale data, it is based on a greedy algorithm. Therefore, the theoretical guarantee is weak and the computational cost is high. In order to solve the SSC problem better, reference [33] propose an iterative weighting (reweighted) l_1 minimisation framework which largely improves the performance of the traditional l_1 minimisation framework—RSSC. In reference [34], they propose a method for controlling the connectivity and sparseness of subspaces by combining the l_1 -norm and l_2 -norm. In [35], a subspace clustering algorithm based on Oracle, named elastic net subspace clustering (EnSC), is proposed. However, this method is sensitive to the initially sparse and connected parameters. Recently, the k-SSC [36] and SR_SSC [37] that reduce the scale of the LASSO problem

from all the datasets to the anchor points set were proposed for selecting k anchor points as support sets.

3. Estimation k and Optimisation of Weight Matrix

Considering that there is an image sequence with F frames, each frame contains the number of β nonrigid motion and P trajectories: $S = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_P\} \in \mathbb{R}^{3F \times P}$, each $\mathcal{S}_i = (X_{1i}, Y_{1i}, Z_{1i}, \dots, X_{Fi}, Y_{Fi}, Z_{Fi})^T$, $i = 1, \dots, P$ is the 3D motion trajectory of the i th point. The process of using the SSC algorithm to segment S can be interpreted as follows: there are β subspaces, and each trajectory \mathcal{S}_i in the sequence can be represented using the sparse linear combination of other trajectories in the same subspace (this can be understood as the same non-rigid object), as: $\mathcal{S}_i = S c_i$. This is illustrated in Figure 1, where each column of S is the trajectory of the 3D point (shown in green). As shown in the visualisation, \mathcal{S}_i can be represented by the linear combination of several other trajectories.

Because of the nonrigidity and overlapping, the sparse subspace algorithm alone cannot segment the multiple non-rigid motions effectively. In Section 2.2, due to the LASSO problem, the general SSC algorithm does not have good expansibility. Therefore, to reduce the computational cost of solving the LASSO problem, we refer to the work in [37] and select a certain number of trajectories (anchor set), $D \in \mathbb{R}^{3F \times k}$, as the support set of each trajectory through hierarchical clustering. Using indices Ω , we express D as follows: $D = S(\cdot, \Omega(j))$, $j = 1, 2, 3, \dots, k$. We replace the self-expression attribute with the following:

$$\mathcal{S}_i = D c_i. \quad (2)$$

In this study, we use the SSC to segment the multiple nonrigid motions in the same scene. Therefore, the SSC model is applied to the multiple nonrigid datasets according to our goal:

$$\min_{C \in \mathbb{R}^{k \times P}} \|C\|_1 + \frac{\mu}{2} \|S - DC\|_F^2 \text{ such that } C_{i,i} = 0 \text{ for } i = 1, 2, 3, \dots, k. \quad (3)$$

Thus, the scale of solving the LASSO problem of the SSC is reduced. Different sizes of datasets are suitable for different k . Unfortunately, in literature [37], k is manually selected, instead of being automatically calculated by the algorithm, according to the different datasets. Apparently, it cannot be effectively applied to real scenes. Therefore, to make k more adaptable to multiple nonrigid motion datasets and improve its robustness, we propose a method to estimate it reasonably and optimise the weight matrix.

3.1. Estimating k . Inspired by reference [38], we attempt to estimate the grouping of data itself using the simple idea of clustering transmission among data and regard it as the corresponding anchor trajectories number k of the dataset.

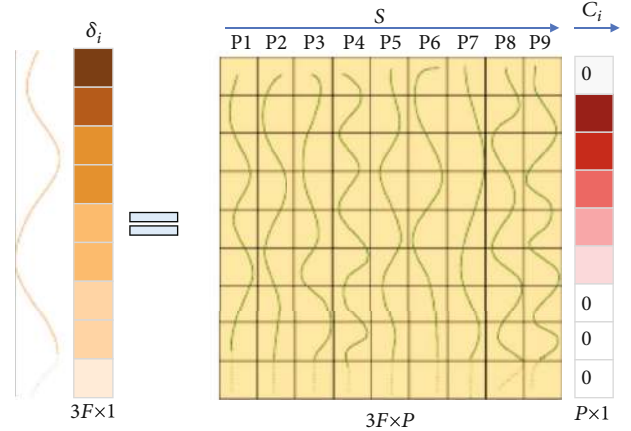


FIGURE 1: The visual representation of the affine subspace constraint, $\mathcal{S}_i = S c_i$, in the trajectory space.

For entire 3D motion trajectories set S , first, calculate the first neighbourhood relationship matrix $\mathcal{K}^1 \in \mathbb{R}^{P \times 1}$ using the accurate distance method. Then, use the first neighbourhood relationship of each trajectory to find the mutual links and groups in all the trajectories. Through the index of the first neighbour of each trajectory, we can directly define an adjacency matrix that is expressed as follows:

$$A(m, n) = \begin{cases} 1, & \text{if } n = \mathcal{K}_m^1 \text{ or } \mathcal{K}_n^1 = m \text{ or } \mathcal{K}_m^1 = \mathcal{K}_n^1, \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

where \mathcal{K}_m^1 is the first neighbour of m , the adjacency matrix connects each trajectory m to its first neighbour through $n = \mathcal{K}_m^1$, forces symmetry through $\mathcal{K}_n^1 = m$, and connects trajectories (m, n) with the same neighbors as $\mathcal{K}_m^1 = \mathcal{K}_n^1$. Equation (4) can return a symmetric sparse matrix and directly specify a connected graph in which the connected components are clusters. Using the undirected graph or directed graph $G = (V, E)$, connected components can be effectively obtained from the adjacency matrix, where V is the set of nodes (clustered trajectories), E is the edge of the connected node, and $A(m, n) = 1$.

The condition in Equation (4) is actually the combination of 1-NN and SNN. This method eliminates the need to solve the problem of graph segmentation. In other words, the adjacency matrix obtained from Equation (4) has absolute connection; thus, it is not necessary to use any distance value as the edge weight, or solve the graph partition.

In paper [38], the main flow of the proposed algorithm is straightforward. After computing the first partition, they proceed to merge these clusters recursively. However, if we set the exact grouping as the number of anchor trajectories k , less error is allowed in the selection of anchor set. The selected trajectories must have good representativeness. However, this is very difficult, because there is no criterion to determine whether the current trajectory can effectively represent the whole sequence.

1. Input: dataset $S^\# \in \mathbb{R}^{P \times 3F}$, where $S^\# = (S)^T$.
2. Output: number of anchor trajectories k .
3. Estimation k :
4. Compute the first adjacency integer matrix $K^1 \in \mathbb{R}^{(P \times 1)}$ using the exact distance method.
5. Given K^1 , obtain ε clusters by using Equation (4) and assume $k = \varepsilon$.

ALGORITHM 1: Estimation k .

1. Input: weight matrix $Z \in \mathbb{R}^{P \times P}$, γ, η, μ .
2. Output: new weight matrix $Z^* \in \mathbb{R}^{P \times P}$.
3. Optimise the weight matrix.
4. Compute the affine matrix W using Equation (6);
5. Initialisation matrix $\mathcal{G} \in \mathbb{R}^{\eta \times P} = 0$;
6. for $i = 1: P$.
7. Compute the γ -neighbors $GN_\gamma(\mathcal{S}_i) = \{\mathcal{S}_{j_\sigma}\}_{\sigma=1}^\gamma$ using Equation (5)
8. num = 1;
9. for $\gamma = 1: \gamma$ do.
10. Compute link $_{ij}$ for $\mathcal{S}_{ij} \in GN_\gamma(\mathcal{S}_i)$ using Equation (7);
11. If (link $_{ij} > \mu$) and (num < η) then
12. $\mathcal{G}_i = \mathcal{G}_i \cup \mathcal{S}_{ij}$;
13. num = num + 1;
14. End if
15. End for
16. If (num < η), then
17. $\mathcal{G}_i = \mathcal{G}_i \cup \{\hat{\mathcal{S}}\}_{\eta-\text{num}}$, where $\{\hat{\mathcal{S}}\}_{\eta-\text{num}}$ have the largest (η -num) similarity rate in $GN_\gamma(\mathcal{S}_i)$;
18. End if
19. End for
20. Compute $Z^* \in \mathbb{R}^{P \times P}$ using Equation (8).

ALGORITHM 2: Optimisation of weight matrix.

Therefore, in this study, by calculating the first adjacency matrix of the sequence once with Equation (4), we obtain a general group ε and designate the number of groups ε the number of anchor trajectories k , like Algorithm 1. This ensures maximum flexibility in the selection of the anchor set. For the convenience of calculation, we use $\mathcal{K}^1 \in \mathbb{R}^{P \times 1}$ when defining the first adjacency matrix; thus, we transpose the sequence S and assume $S^\# = (S)^T$.

3.2. Optimisation of Weight Matrix. After determining the values of k , we use the hierarchical clustering based on the factorisation of the rank-two nonnegative matrix given in [39] to select the set of anchor trajectories. Although this method is somewhat random, it has been proven in literature [37] that this method is better than the random uniform selection used by SSSC [30]. However, it is very difficult to select the anchor trajectories without any prior conditions. Although the author has provided theoretical proof for this method in paper [37], the randomness in the algorithm cannot be ignored. To reduce the decisive influence of the anchor trajectories selection on the algorithm's effectiveness, the idea of multilayer graphs is added in reference [37]. In our subsequent experiments, it proves that in multiple nonrigid scenes,

the multilayer graphs alone cannot improve the effect of bad selection. For this reason, we refer to [12], which uses a post-processing technology, and modifies the weight matrix through the good neighbors to optimise the sparseness and connectivity, the importance of the anchor trajectories is minimised.

$$GN_\gamma(\mathcal{S}_i) = \{\mathcal{S}_j\}_{j=1}^\gamma = \underset{x_j}{\operatorname{argmax}} \sum_{j=1}^\gamma |w_{ij}|. \quad (5)$$

For each trajectory, \mathcal{S}_i , we select γ , the most similar trajectories from the affine matrix W , using Equation (5), while $\gamma < k$, $(\mathcal{S}_i, \mathcal{S}_j) \in S$, $w_{ij} \in W$, and W :

$$W = \frac{1}{2} (|Z| + |Z^T|). \quad (6)$$

where $|\cdot|$ is the absolute value of matrix element, $Z \in \mathbb{R}^{P \times P}$, $Z(\Omega(j), :) = C(j, :)$ for $j = 1, 2, \dots, k$, and the other rows of Z are equal to zero. To better reflect the strong connection of the trajectories in the same subspace and the weak

connection of trajectories across different subspaces, the following conditions are introduced:

$$\text{link}_{ij} = \sum_{\sigma=1}^{\gamma} 1, \text{ for } 1 = \begin{cases} 1, \mathcal{S}_i \in GN_{\gamma}(\mathcal{S}_{j_{\sigma}}), \\ 0, \text{ otherwise.} \end{cases} \quad (7)$$

Equation (7) can be understood as follows. In the dataset, S , for each \mathcal{S}_i , we have $\mathcal{S}_j \in GN_{\gamma}(\mathcal{S}_i)$ and $GN_{\gamma}(\mathcal{S}_j) = \{\mathcal{S}_{j_{\sigma}}\}_{\sigma=1}^{\gamma}$; thus, there are link_{ij} common neighbors between \mathcal{S}_i and \mathcal{S}_j . We take the link_{ij} as the judging condition; when $\text{link}_{ij} > \mu$, we assume that \mathcal{S}_j is one of the good neighbors of \mathcal{S}_i . In terms of calculation efficiency, due to the increased path length, the relationship between samples will be weaker. Therefore, to ensure the calculation efficiency, we assume $\mu = 1$ in the rest of this study; that is, when there is more than one common similarity between $(\mathcal{S}_i, \mathcal{S}_j)$, \mathcal{S}_j can be referred to as the good neighbour of \mathcal{S}_i . Finally, for each \mathcal{S}_i , we extract η good neighbors from $GN_{\gamma}(\mathcal{S}_i)$ and express the set of good neighbors as follows: $\mathcal{E} \in \mathbb{R}^{\eta \times P}$, where $\eta \leq \gamma$. Finally, to reduce the error of spectral clustering, we limit: $0 \leq z_{ij}^* < 1$ and set the sum of each row of optimised Z^* as 1:

$$z_{ij}^* = \begin{cases} \frac{w_{ij}}{\sum_j w_{ij}}, & \text{if } \mathcal{S}_j \in \mathcal{E}_i; \\ 0, & \text{if } \mathcal{S}_j \notin \mathcal{E}_i, \end{cases} \quad (8)$$

where $\sum_j w_{ij}$ is the sum of w_{ij} between \mathcal{S}_i and $\{\mathcal{S}_j\}_{j=1}^{\eta}$ and $\{\mathcal{S}_j\}_{j=1}^{\eta} \in \mathcal{E}_i$.

4. Proposed Method

In the third section, we analyse the process of estimation based on the number of anchor trajectories k and weight matrix optimisation. Then, we integrate the theory and build a complete multiple nonrigid segmentation framework. Based on the idea of multilayer graphs [11], we choose the L group, each group contains k anchor trajectories, using indices $\Omega^{(i)} (i = 1, 2, \dots, L)$, express the dictionary dataset of group L as $\{D^{(i)} = S(:, \Omega^{(i)}(j)) \in \mathbb{R}^{3F \times k}\}_{i=1}^L, j = 1, 2, 3, \dots, k$. Similarly, $\{C^{(i)} \in \mathbb{R}^{k \times P}\}_{i=1}^L$ is the weight matrix C corresponding to a L -level graph. The optimisation problem (3) can be converted to the following equation:

$$\min_{C^{(i)} \in \mathbb{R}^{k \times P}} \left\| C^{(i)} \right\|_1 + \frac{\mu}{2} \left\| S - D^{(i)} C^{(i)} \right\|_F^2, C_{j, \Omega^{(i)}(j)}^{(i)} = 0 \text{ for } j = 1, 2, \dots, k. \quad (9)$$

For optimisation problem (9), we can use the alternating direction method of multipliers (ADMM) algorithm to optimise [37, 40]. Considering our requirements, the ADMM algorithm is very suitable because it has a low computational cost per iteration (linear in the number of variables). Furthermore, its slow convergence (linear at best) is not a bottleneck, because a high precision is not necessary as we only need to

know the order of magnitude of the entries of $C^{(i)}$. Moreover, it is not necessary to solve (9) with high precision, because the data is usually very noisy.

Then, when we obtain $(C^{(i)})_{i=1}^L$, we build a L -level graph $G, G^{(i)} = (V, W^{(i)})$, where the trajectories to be clustered are in V , and $W^{(i)}$ is the affine matrix of i th layer. In order to calculate the affine matrix later, and at the same time to distinguish easily, we redefine the weight matrix, $Z^{(i)} \in \mathbb{R}^{P \times P} = 0$:

$$Z^{(i)} \left(\Omega^{(i)}(j), : \right) = C^{(i)}(j, :) \text{ for } j = 1, 2, \dots, k \quad (10)$$

We use Algorithm 2 to optimise the weight matrix $Z^{(i)}$ for the i th layer, making the affinity coefficient more representative. Then, $W^{(i)}$ can be obtained as follows:

$$W^{(i)} = \frac{1}{2} \left(\left| Z^{(i)} \right| + \left| Z^{(i)T} \right| \right). \quad (11)$$

Following the spectral clustering [9], we compute the Laplacian matrix $P^{(i)}$ of each layer as follows:

$$P^{(i)} = I_E - D_g^{(i)-1/2} W^{(i)} D_g^{(i)-1/2}, \quad (12)$$

where I_E is the identity matrix and $D_g^{(i)}$ is the diagonal matrix, whose value is the sum of each row of $W^{(i)}$ is expressed as follows:

$$D^{(i)}(i, j) = \sum_{j=1}^P W^{(i)}(j, :). \quad (13)$$

According to the derivation of reference [11], we integrate the multilayer graphs. For each layer of graph, $G^{(i)}$, it is necessary to obtain its corresponding β -dimensional subspace representation matrix $\Gamma^{(i)} \in \mathbb{R}^{P \times \beta}$, which is composed of the eigenvectors corresponding to the β smallest eigenvalues in $P^{(i)}$. Finally, the information of the graph, G , is integrated to obtain the expression of the final Laplace matrix P_f :

$$P_f = \sum_{i=1}^L P^{(i)} - \alpha \sum_{i=1}^L \Gamma^{(i)} \left(\Gamma^{(i)} \right)^T. \quad (14)$$

By using the k -means to classify $\Gamma \in \mathbb{R}^{P \times \beta}$, the eigenvector corresponding to the β smallest eigenvalue of the Laplacian matrix P_f , we can obtain the final classification of S .

5. Experiment

5.1. Data and Operating Environment. Most of the sparse datasets used in this study are from <http://cvlab.lums.edu.pk/nrsfm>; the others are from some open-source projects. We make reference to Kumar et al. [15, 16] and merge

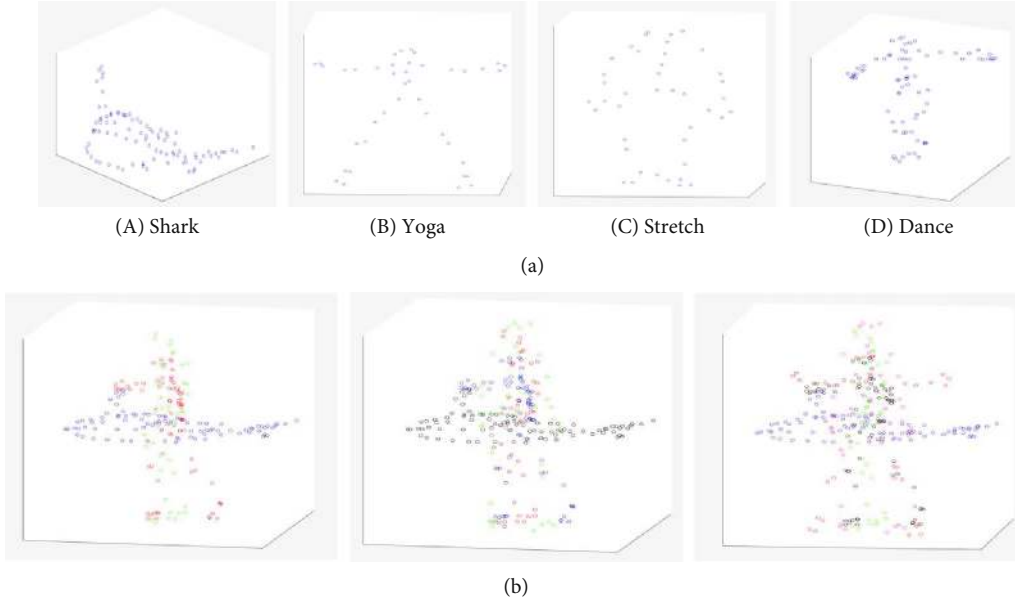


FIGURE 2: (a) Visualisation of some NRSFM datasets. (b) Example of synthetic datasets reference in [15, 16]. Different colors represent different motion.

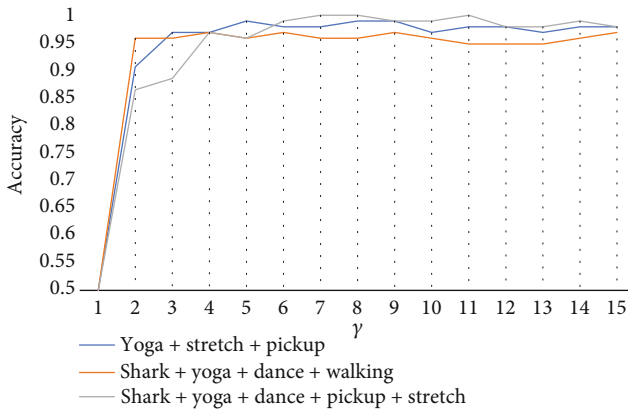


FIGURE 3: The x -axis represents the value of γ , the y -axis represents the accuracy of segmentation, and the three-colour line graph represents the experiment on three different datasets.

multiple datasets to simulate multiple nonrigid motions on the same scene, as shown in Figure 2(b). In Figure 2(b), different movements are represented by different color blocks. The sparse NRSFM datasets that we used are as follows: dance (75 feature points), shark (91 feature points), walk (55 feature points), yoga (41 feature points), stretch (41 feature points), face (37 feature points), pickup (41 feature points), and drink (41 feature points), some show in Figure 2(a). The operating system used in this experiment is macOS, which is configured as 3.1 GHz Intel Core i5. The memory is 8 GB, and the MATLAB version is 2017b.

5.2. Performance Evaluation Index. To better compare the difference between the segmentation results of the algorithm, we measure the segmentation effect using two indicators: the correct rate and the number of incorrectly classified trajectories.

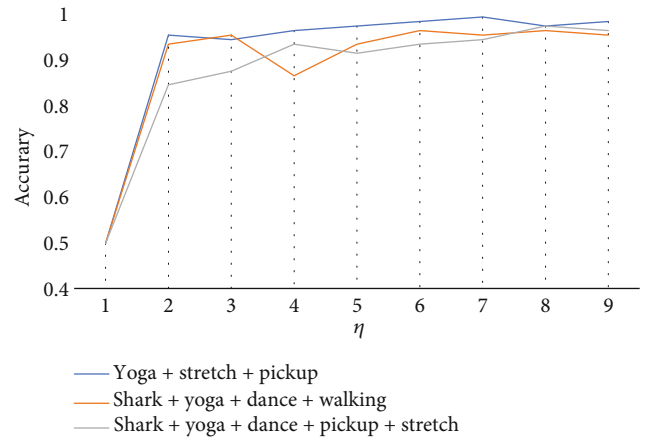


FIGURE 4: The x -axis represents the value of η , the y -axis represents the corresponding segmentation accuracy, and the three-colour line graph represents the three datasets.

Accuracy rate:

$$R_a = \frac{\# \text{of correctly classified trajectories}}{\text{total} \# \text{of trajectories}}. \quad (15)$$

Errors(n): the number of incorrectly classified trajectories is the algorithm segmentation errors, defined as follows:

$$\text{Error}(n) = \# \text{of incorrectly classified trajectories}. \quad (16)$$

5.3. Experimental Parameter. When the algorithm is running, three parameters are required: α , γ , and η .

α : in reference [11], a parameter, α , is required to balance the connectivity between each layer of graphs and the representation of the distance between subspaces, that is, Equation (14). In [11], the author proposes setting it to 0.5 and states

TABLE 1: Analysis of the anchor trajectories number k estimation.

Dataset	SR_SSC (60)	SR_SSC (80)	SR_SSC (100)	SR_SSC (Section 3.1)	MNR_S
	R_α	R_α	R_α	R_α	R_α
Shark+dance+yoga (207)	0.985	0.986	<i>0.990</i>	0.988 (54)	0.993
Yoga+stretch+pickup (123)	0.967	0.945	0.926	<i>0.974 (36)</i>	0.980
Dance+pickup+stretch(157)	0.956	0.951	0.921	<i>0.976 (41)</i>	0.990
Shark+dance+yoga+walking (262)	0.961	0.978	0.983	0.969 (69)	<i>0.981</i>
Shark+pickup+stretch+dance (248)	0.907	0.902	<i>0.943</i>	0.920 (67)	0.970
Shark+dance+yoga+pickup+stretch (289)	0.730	0.742	<i>0.768</i>	0.750 (78)	0.930
Dance+walking+shark+yoga+stretch (303)	0.789	0.802	<i>0.810</i>	0.809 (82)	0.963

1. Input: 3D feature matrix $S \in \mathbb{R}^{3F \times P}$, where F is the frame, and P is the number of feature points, and the number of multiple non-rigid β (as number of layers L at the same time), α, γ, η .
2. Output: a clustering of the columns of S in subspaces
3. Segmentation
4. Computation of k using Algorithm 1 to S^T .
5. for $i = 1: \beta$ do
6. Obtain dictionary $D^{(i)} \in \mathbb{R}^{3F \times k}$ by using random hierarchical clustering [2]
7. Compute $C^{(i)}$ using ADMM to solve (9).
8. Compute $Z^{(i)}$ using Equation (10).
9. Optimise $Z^{(i)}$ using Algorithm 2.
10. Compute $W^{(i)}$ using Equation (1).
11. Compute $P^{(i)}$ using Equation (12).
12. Compute $\Gamma^{(i)} \in \mathbb{R}^{P \times \beta}$.
13. End for
14. Compute P_f using Equation (14).
15. Compute the $\Gamma \in \mathbb{R}^{P \times \beta}$.
16. The final clustering of the columns of S is obtained by clustering the rows of Γ using the k -means with β clusters.

ALGORITHM 3: Multibody NRSFM segmentation

that the theory of multilayer graphs is not sensitive to the setting of α ; at the same time, we need the contrast algorithm, SR_SSC, to set α to 0.5. To facilitate the comparison between the algorithms, we set α to be equal to 0.5

γ : in Section 3.2, when optimizing the weight matrix, for each data point, it is necessary to select the largest similarity points. To explain the influence of the γ setting on the algorithm, we observed the effect of varying γ on the algorithm. We chose three datasets of different sizes: (1) yoga+stretch+pickup, (2) shark+yoga+dance+walking, and (3) shark+yoga+stretch+pickup+dance. Furthermore, we will assume $\eta = \gamma$ (in other words, $GN_\gamma = \mathcal{E}$). Each dataset was tested 20 times to calculate the average accuracy. The results are shown in Figure 3. From the trace of the curve in Figure 3, it can be observed that after $\gamma \geq 6$ in the three datasets, the segmentation accuracy tends to stabilise. Therefore, to improve the scalability of the algorithm, in this study, we set γ as 10.

η : similarly, to illustrate the influence of η on the algorithm, 20 experiments were conducted using the same three datasets. Here, the value of γ is 10, and the average accuracy is calculated, as shown in Figure 4. From the trend of the curve in the figure, it can be seen that the assumption that η is 7 is relatively reasonable.

5.4. Analysis of the Anchor Trajectories Number k Estimation.

To analyse the rationality of the estimation of k by Algorithm 1 in Section 3.1, we construct five composite data, as shown in Table 1. Because our Algorithm 3 refers to the basic idea of the SR_SSC [37], Table 1 only compares the experimental results of our Algorithm 3, namely, multibody non-rigid segmentation (MNR_S) and the SR_SSC algorithm using different k . The code of the SR_SSC provided open-source links in the relative literature.

To mitigate the effect of the randomness of the anchor trajectories selection on the reliability of the results, we conducted 20 experiments on each dataset to calculate the average accuracy, and the results are shown in Table 1. In Table 1, bold indicates the best effect on the dataset, followed by italics. For comparison, all the parameters, with the exception of the anchor trajectories and k , are set to be constant. By setting the k in the SR_SSC algorithm as 60, 80, and 100, respectively, and estimating the formula in Section.3.1, four situations are obtained. In Table 1, in the section of SR_SSC (Section 3.1), k , as obtained by the estimation of Algorithm 1, is in parenthesis beside the accuracy value, and the numbers in parentheses in the dataset represent the total number of feature points in the dataset.

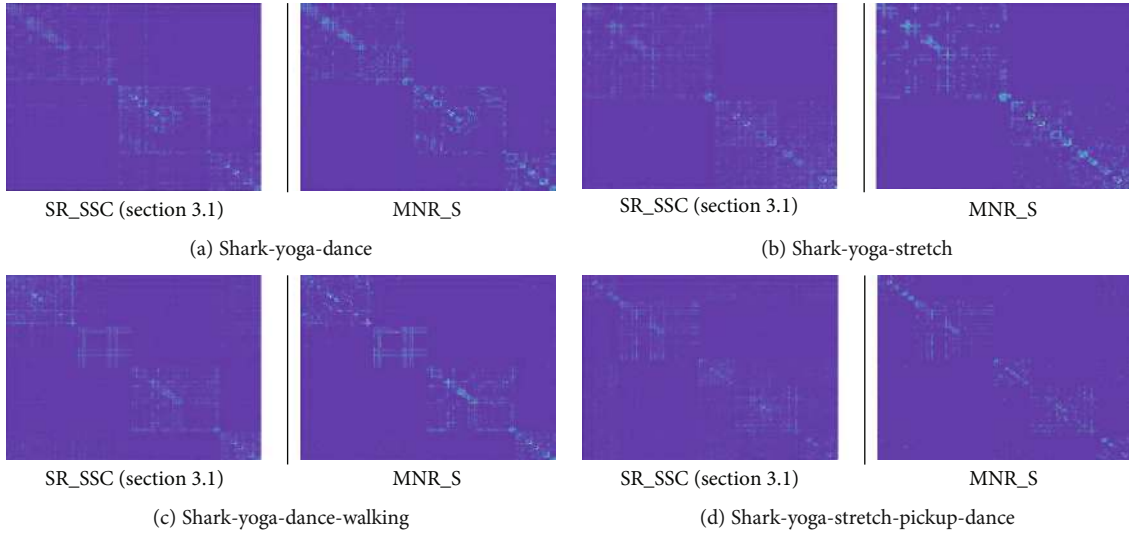


FIGURE 5: The affine matrix obtained by our algorithm and SR_SSC (Section 3.1).

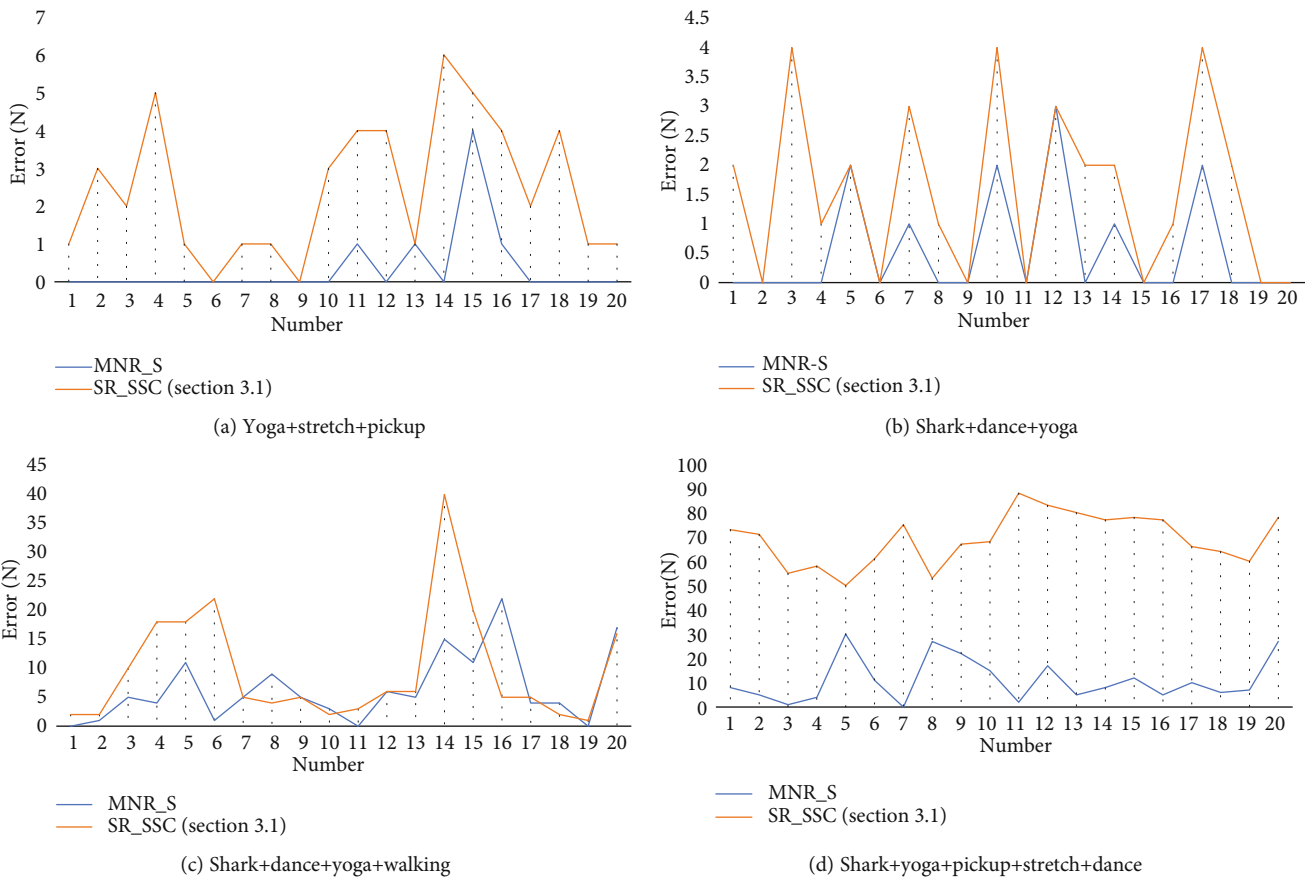


FIGURE 6: Influence of optimisation weight matrix on algorithm performance and comparison of the SR_SSC (Section 3.1) algorithm and ours in 20 experiments on each flow.

From Table 1, we can see that the appropriate anchor trajectories of SR_SSC are different for the different data scales. Using the method of estimating k that we introduced in Section 3.1, the experimental result shows that the results

are relatively good across different datasets. Moreover, after combining the good neighbors in our method, our multi-body nonrigid segmentation(MNR_S) is greatly improved, and the average accuracy is more than 90%.

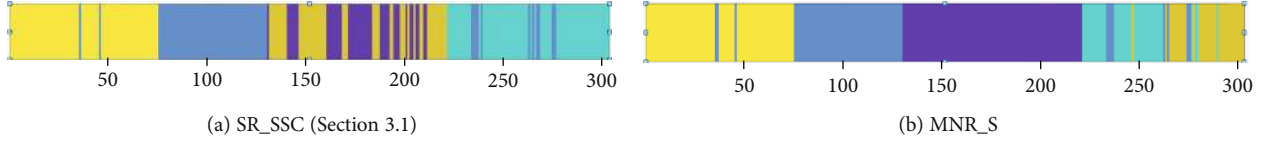


FIGURE 7: Clustering result of ‘dance+walking+shark+yoga+stretch.’.

TABLE 2: Accuracy ($R_{(a)}$) of five algorithms in the 15 datasets synthesised.

	Dataset	SSC	EnSC	k-SSC	SSSC	MNR_S
Three objects	Shark+dance+yoga (207)	0.992	1.000	0.903	0.928	0.993
	Yoga+stretch+pickup (123)	0.919	0.959	0.920	0.903	0.986
	Dance+pickup+stretch (157)	0.828	0.942	0.940	0.890	0.987
	Shark+walking+drink (187)	1.000	0.983	0.968	0.902	0.989
	Dance+yoga+drink (157)	0.917	0.920	0.902	0.886	0.980
Four objects	Shark+dance+yoga+walking (262)	1.000	0.996	0.901	0.907	0.980
	Dance+stretch+pickup+shark (248)	0.714	0.810	0.779	0.800	0.968
	Walking+face+pickup+drink (174)	0.701	0.850	---	0.813	0.948
	Dance+yoga+drink+walking (212)	0.755	0.780	---	0.834	0.984
	Dance+yoga+stretch+pickup (198)	0.858	0.803	0.737	0.785	0.989
Five objects	Shark+dance+yoga+pickup+walking(303)	0.736	0.799	0.742	0.783	0.939
	Shark+yoga+pickup+stretch+dance (289)	0.773	0.775	---	0.757	0.980
	Shark+yoga+walking+drink+dance (303)	0.731	0.791	0.751	0.698	0.970
	Dance+yoga+stretch+pickup+walking (253)	0.731	0.704	0.681	0.708	0.964
	Drink+walking+yoga+pickup+stretch(219)	0.739	0.794	---	0.750	0.937

5.5. *Analysis of the Influence of Optimisation Weight Matrix on Algorithm.* To observe the effect of optimising the weight matrix in addition to using the multilayer graph method, we visualized the difference of affinity matrices obtained by our method MNR_S and SR_SSC (Section 3.1) in Figure 5. Comparing the error correction ability of the same set of anchor trajectories separately, draw Error(n) in 20 experiments on the SR_SSC and our algorithm, as shown in Figure 6. Furthermore, we show the clustering results for one dataset, ‘Dance+walking+shark+yoga+stretch’ in the same group of experiments, as shown in Figure 7. In Figure 7, the same color indicates belonging to the same motion, small blocks in different colors mean clustering error, and abscissa is expressed as feature point. Because each motion contains different feature points, so the color block width displayed is different.

In Figure 5, it can be observed that the affine matrix calculated in our algorithm effectively removes the interference items around, ensures the more representative affinity coefficient is concentrated near the main diagonal, reduces the influence of interference coefficient on segmentation, and makes the algorithm more stable.

From the curve trend of Figure 6, that in the multiple nonrigid scenes, our algorithm is obviously superior in terms of stability and error rate. Especially in the ‘shark+dance+yoga+stretch + pickup’ and ‘shark+dance+yoga+walking’ datasets, the error rate is not ideal under the influence of high motion similarity when only the multilayer graphs is used. In

TABLE 3: The effect of reconstruction using BMM after segmentation in two objects.

Dataset	BMM+SSC (2D)	BMM+MNR_S (2D)
	e_{3D}	
Dance+yoga	0.170	0.150
Drink+walking	0.085	0.078
Pickup+yoga	0.185	0.151
Drink+stretch	0.092	0.073
Shark+dance	0.201	0.171

contrast, our algorithm cooperates with the optimisation of the weight matrix, reduces the interference term of the final obtained affine matrix, and visibly reduces the error.

5.6. *Comparison with Other SSC Algorithms on Multibody NRSFM.* In this subsection, we compare the segmentation results of other SSC algorithms to demonstrate the stability of our algorithm in multibody NRSFM segmentation. We compare our algorithm MNR_S with the SSC, SSSC [30], EnSC [35], and k_SSC [36] algorithms on the synthesised dataset. To rule out accidents, 20 experiments are conducted on the same set of data, and the average accuracy is obtained. The results are shown in Table 2. To ensure fairness, the initial parameters in the EnSC algorithm are different in each dataset. For each dataset, we debug a variety of possible parameter sets for the EnSC algorithm to achieve the best

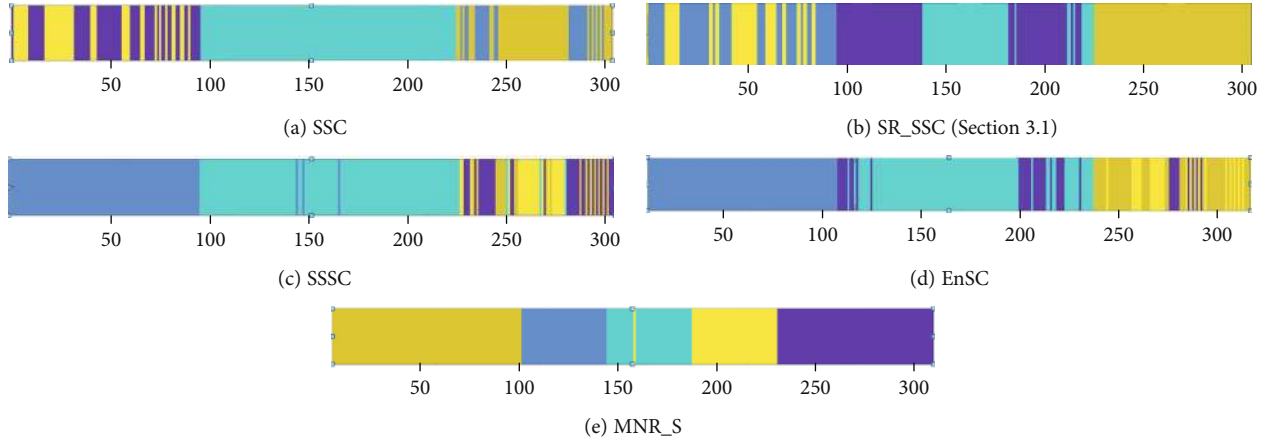


FIGURE 8: Clustering result of ‘shark+yoga+pickup+stretch+dance.’.

effect in the current experimental dataset. The dictionary set of the SSSC, with the size of 30β , adopts the random uniform selection method, where β is the number of nonrigid bodies. (Remarks: in Table 2, bold indicates the best effect on the dataset, followed by italics; ‘—’ indicates that the segmentation effectiveness is still less than 50% after adjusting the parameters.)

As can be seen from Table 2, in general, the segmentation accuracy of our algorithm is above 90%. However, the effectiveness of other algorithms is sometimes lower than or more than 90%.

For the three nonrigid datasets, the five algorithms perform relatively well. Although the EnSC algorithm achieves 100% segmentation accuracy on the shark+dance+yoga dataset, and the SSC achieves 100% segmentation accuracy on the ‘shark+walking+drink’ dataset. In contrast, our algorithm is very stable on five synthetic datasets, with an average segmentation accuracy of 98%.

For the four nonrigid scenes, the EnSC and SSC algorithms achieve good results on the ‘shark+dance+yoga+walking’ dataset; however, the accuracy of the remaining four datasets is not as high as 90%. The average accuracy of our algorithm in five datasets exceeds 94%.

For the five nonrigid scenes, our algorithm’s accuracy is also more than 90%; the other algorithms are not ideal (for the k-SSC, we also attempt to adjust the various parameters and observe the effect. However, many synthetic data remain unideal).

In Table 3, for two objects in the scene, SSC and MNR_S are used to segment the 2D observation matrix first, and then, the BMM is used for reconstruction. Because our segmentation algorithm can segment the scene of two objects very well, the effect is significantly better than that of SSC+BMM(e_{3D} is expressed as the 3D reconstruction error calculated by BMM).

We also extract a dataset, ‘shark+yoga+stretch+pickup+dance,’ and show four comparison algorithms and our method’s segmentation result in Figure 8, in which different colors represent different motions, small blocks in different colors mean clustering error. It can be seen that the segmentation result of our method is obviously better. The other algo-

gorithms cannot segment the middle three motions because of the high similarity, instead, our algorithm is very ideal.

MNR_S is less sensitive to the parameters. For example, from Section 5.3, we can observe that the fluctuation of the average accuracy rate when the value of γ exceeds 3 is minimal. Likewise, for η , varying the initial parameters in the normal range has little effect on our algorithm. At the same time, we use the combination of the good neighbour and the multilayer graphs to eliminate the influence of the poor selection of anchor set on the algorithm. The experimental results demonstrate that our algorithm is stable and performs robustly on our synthetic dataset.

6. Conclusion

The main purpose of this study is to deepen the research on ‘3D reconstruction and segmentation of multi-body NRSFM.’ To study the segmentation effect of current subspace clustering algorithms on multiple nonrigid bodies on the same scene, we drew our inspiration from Abdullali et al.’s study [37]. At the same time, we realised that because the number of anchor trajectories in the SR_SSC algorithm was manually determined, it could not be effectively adapted to all datasets; and the segmentation has certain randomness. With the first neighbour of each trajectory, we use the concept of hierarchical clustering to mine the grouping between data, estimate the number of anchor trajectories k , and selected anchor trajectories by random hierarchical clustering. Then, we modify the weight matrix obtained by good neighbors to improve the affinity coefficient. At present, from the perspective of the experiments, our algorithm is effective for the segmentation of multibody NRSFM. Furthermore, the proposed algorithm can be more suitable for different scenes of multibody NRSFM, because the initial parameters have little influence on it.

Data Availability

The data used to support the study can be available upon request

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Natural Science Foundation of Zhejiang Province (LZ20F020003, LY17F020034, and LSZ19F010001), the National Natural Science Foundation of China (61272311, 61672466), and the 521 Project of Zhejiang Sci-Tech University.

References

- [1] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, "Nonrigid structure from motion in trajectory space," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 41–48, 2008.
- [2] Y. Dai, H. Li, and M. He, "A simple prior-free method for non-rigid structure-from-motion factorization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2018–2025, RI, USA, 2012.
- [3] A. Agudo and F. Moreno-Noguer, "A scalable, efficient, and accurate solution to non-rigid structure from motion," *Computer Vision and Image Understanding*, vol. 167, no. 2, pp. 121–133, 2018.
- [4] S. H. N. Jensen, M. E. B. Doest, H. Aanæs, and A. Del Bue, "A benchmark and evaluation of non-rigid structure from motion," *International Journal of Computer Vision*, 2020.
- [5] J. Liang, J. Yang, M.-M. Cheng, P. L. Rosin, and L. Wang, "Simultaneous subspace clustering and cluster number estimation based on triplet relationship," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 3973–3985, 2019.
- [6] C. You and R. Vidal, "Geometric conditions for subspace-sparse recovery," in *In Proceedings of the 32nd International Conference on Machine Learning*, pp. 1585–1593, Lille, 2015.
- [7] L. Zappella, A. Del Bue, X. Llado, and J. Salvi, "Joint estimation of segmentation and structure from motion," *Computer Vision and Image Understanding*, vol. 117, no. 2, pp. 113–129, 2013.
- [8] S. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation in the presence of outlying, incomplete or corrupted trajectories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1832–1845, 2010.
- [9] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2001.
- [10] X. Zhu, S. Zhang, Y. Li, and J. Zhang, "Low-rank sparse subspace for spectral clustering," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–12, 2018.
- [11] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering on multi-layer graphs via subspace analysis on Grassmann manifolds," *IEEE Transactions on Signal Processing*, vol. 62, no. 4, pp. 905–918, 2014.
- [12] J. Yang, J. Liang, K. Wang, P. L. Rosin, and M.-H. Yang, "Subspace clustering via good neighbors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 1537–1544, 2019.
- [13] C. Russell, R. Yu, and L. Agapito, "Video pop-up: monocular 3d reconstruction of dynamic scenes," in *European Conference on Computer Vision*, pp. 583–598, Springer, 2014.
- [14] Y. Zhu, D. Huang, F. De La Torre, and S. Lucey, "Complex non-rigid motion 3D reconstruction by union of subspaces," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1542–1549, Columbus, OH, USA, 2014.
- [15] S. Kumar, Y. Dai, and H. Li, "Spatio-temporal union of subspaces for multi-body non-rigid structure-from-motion," *Pattern Recognition*, vol. 71, pp. 428–443, 2017.
- [16] S. Kumar, Y. Dai, and H. Li, "Multi-body non-rigid structure-from-motion," in *2016 Fourth International Conference on 3D Vision (3DV)*, pp. 148–156, Stanford, CA, USA, 2016.
- [17] M. Rocchetti, G. Delnevo, L. Casini, and P. Salomoni, "A cautionary tale for machine learning design: why we still need human-assisted big data analysis," *Mobile Networks and Applications*, vol. 25, no. 3, pp. 1075–1083, 2020.
- [18] G. Delnevo, M. Rocchetti, and S. Mirri, "Intelligent and good machines? The role of domain and context codification," *Mobile Networks and Applications*, vol. 25, pp. 977–985, 2020.
- [19] C. Kong and S. Lucey, "Deep non-rigid structure from motion," in *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1558–1567, Seoul, South Korea, 2019.
- [20] I. T. Jolliffe, "Principal component analysis and factor analysis," in *Principal Component Analysis*, pp. 115–128, Springer, 1986.
- [21] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1945–1959, 2005.
- [22] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.
- [23] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.
- [24] S. Xiao, W. Li, D. Xu, and D. Tao, "FaLRR: a fast low rank representation solver," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4612–4620, Boston, MA, USA, 2015.
- [25] G. Liu, H. Xu, J. Tang, Q. Liu, and S. Yan, "A deterministic analysis for LRR," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 3, pp. 417–430, 2016.
- [26] C. Y. Lu, H. Min, Z. Q. Zhao, L. Zhu, D. S. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *European Conference on Computer Vision*, pp. 347–360, Springer, 2012.
- [27] E. Elhamifar and R. Vidal, "Sparse subspace clustering: algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.
- [28] M. Soltanolkotabi, E. Elhamifar, and E. J. Candès, "Robust subspace clustering," *Annals of Statistics*, vol. 42, no. 2, pp. 669–699, 2014.
- [29] M. Soltanolkotabi and E. J. Candès, "A geometric analysis of subspace clustering with outliers," *Annals of Statistics*, vol. 40, no. 4, pp. 2195–2238, 2012.
- [30] X. Peng, L. Zhang, and Z. Yi, "Scalable sparse subspace clustering," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 430–437, Portland, OR, USA, 2013.
- [31] C. You, D. P. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3918–3927, Las Vegas, NV, USA, 2016.
- [32] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE*

- Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.
- [33] J. Xu, K. Xu, K. Chen, and J. Ruan, “Reweighted sparse subspace clustering,” *Computer Vision and Image Understanding*, vol. 138, pp. 25–37, 2015.
 - [34] Y. Panagakis and C. Kotropoulos, “Elastic net subspace clustering applied to pop/rock music structure analysis,” *Pattern Recognition Letters*, vol. 38, pp. 46–53, 2014.
 - [35] C. You, C.-G. Li, D. P. Robinson, and R. Vidal, “Oracle-based active set algorithm for scalable elastic net subspace clustering,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3928–3937, Las Vegas, NV, USA, 2016.
 - [36] S. Tierney, Y. Guo, and J. Gao, “Efficient sparse subspace clustering by nearest neighbour filtering,” 2017, <http://arxiv.org/abs/1704.03958v1arXiv>.
 - [37] M. Abdolali, N. Gillis, and M. Rahmati, “Scalable and robust sparse subspace clustering using randomized clustering and multilayer graphs,” *Signal Processing*, vol. 63, pp. 166–180, 2019.
 - [38] S. Sarfraz, V. Sharma, and R. Stiefelhagen, “Efficient parameter-free clustering using first neighbor relations,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8926–8935, Long Beach, CA, USA, 2019.
 - [39] N. Gillis, D. Kuang, and H. Park, “Hierarchical clustering of hyperspectral images using rank-two nonnegative matrix factorization,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 4, pp. 2066–2078, 2015.
 - [40] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.