Point Cloud Segmentation based on Hypergraph Spectral Clustering

Songyang Zhang, Shuguang Cui, and Zhi Ding
University of California, Davis

Abstract. Hypergraph spectral analysis has become a significant tool in data analysis due to its power in processing complex data structure. In this work, we investigate its efficacy for 3D point cloud segmentation, where the underlying surface of point clouds and the multilateral relationship among the points can be naturally captured by the high-dimensional hyperedges. Leveraging the power of hypergraph signal processing, we estimate and order the hypergraph spectrum from observed coordinates. By trimming redundant information from the estimated hypergraph spectral space based on the strengths of spectral components, we develop a clustering-based method to achieve effective segmentation. We apply the proposed segmentation method to different point clouds, and analyze their respective spectral properties. Our experimental results clearly demonstrate the effectiveness of estimated hypergraph spectrum and efficiency of the proposed segmentation method.

Keywords: Hypergraph, point cloud, signal processing, spectral clustering, segmentation.

1 Introduction

With the proliferation of low-cost sensing hardware, three-dimensional (3D) point clouds have been widely adopted as an efficient representation for 3D objects and their surroundings in many applications [1]. One of the essential processing tasks on 3D point clouds is segmentation. Similar to image segmentation, the goal of point cloud segmentation is to identify and cluster the points in a cloud with similar features into their respective regions [2]. These partitioned region should be physically meaningful. For example, the authors of [3] segmented the human posture point clouds to analyze human behaviors, where the human body is partitioned into clusters to represents different semantic body parts. The segmentation can help analyze the point clouds in various applications, such as object tracking, object classification, feature extraction, and feature detection.

To establish a general model and segment efficiently in the point clouds, we consider a new high-dimensional graph representation known as hypergraphs. A hypergraph \( \mathcal{H} = (\mathcal{V}, \mathcal{E}) \) is the extension of traditional graph, consisting of nodes \( \mathcal{V} \) and hyperedges \( \mathcal{E} \). Generalizing edges from traditional graphs, each hyperedge can connect more than two nodes, i.e., a subset of \( \mathcal{V} \). Since hypergraphs generalize graphs, a graph is only a special case of hypergraph, in which each hyperedge
Fig. 1. Point Clouds with Geometric Model. (a) 7 points sampled from a cylinder, where green nodes are from the top, blue nodes are from the boundaries, black nodes are from the bottom, and brown nodes are from the side. (b) A graph model of the cylinder samples based on distances with 7 nodes and 8 edges. (c) A hypergraph model of cylinder samples with 3 hyperedges, where each hyperedge connects the nodes from the same surfaces. (d) A point clouds consisting of three square surfaces. (e) A sampled point cloud via hypergraph harr-like high pass filter [4].

connects exactly two nodes. The high-dimensionality of hyperedges can directly model multilateral relationship of points over the point cloud surfaces, thereby making hypergraph a convenient model for point clouds. Fig.1(a)-(c) show typical examples of graph and hypergraph models, where the graph model cannot distinguish different surfaces but the hypergraph model can capture the surface information.

Moreover, in hypergraph signal processing (HGSP), hypergraph spectrum has shown great promise in tasks like clustering and compression [5]. The hypergraph frequency can be used to analyze the underlying structure of datasets. For example, a hypergraph Harr-like high frequency filter can extract sharp features in point clouds, as illustrated in Fig.1(e). Major challenges of applying HGSP in point clouds are the construction of hypergraph structure and the efficient computation of the hypergraph spectral space. One traditional approach to derive the hypergraph spectrum is to first construct the hypergraph based on distances among nodes before decomposing the resulting (adjacency/Laplacian) tensor. However, such distance-based hypergraph construction is deficient in measuring efficiency. Furthermore, the decomposition of the representing tensor can also be time-consuming.

To overcome the aforementioned challenges, we propose a novel method of spectral clustering segmentation based on the hypergraph signal processing for the gray-scale point clouds. To construct the hypergraph and estimate the spectral space efficiently, we first develop a method to estimate spectral components based on hypergraph stationary process before ordering the components according to their frequency coefficients. Removing redundant information based on the spectrum order, a spectral clustering can be implemented on the key spectrum components to segment the point clouds. We test the proposed segmentation methods on multiple gray-scale point cloud datasets. The test performance of the proposed method validates its effectiveness and efficiency. Note that, the
proposed hypergraph construction can be applied to applications beyond point cloud segmentation. We shall point out many future directions of HGSP on point clouds, such as compression and denoising.

2 Related Works

In this section, we overview the related research works on point cloud segmentation and hypergraph signal processing.

Segmentation over 3D Point Clouds: There are a plethora of methods proposed to segment point clouds efficiently. For example, edge-based methods, such as [6], detect boundaries of regions to segment point clouds and facilitate fast implementation but can be sensitive to noise. Another widely-used class of segmentation methods are region-based [7,8], and are more robust to noises but suffer from either over or under segmentation.

In addition, attribute-driven approaches based on clustering, are also used in segmentation [9,10]. These attribute-based methods would first compute attributes before clustering the derived attributes. Their performances depend highly on the quality of the derived attributes. Thus, how to accurately compute attributes poses a problem. Fortunately, graphs have been successfully applied to extract the underlying structural information of point clouds. Modeling point clouds as graphs, we can derive graph spectral space and utilize spectral clustering to achieve segmentation. Generally, graph-based methods deliver good results and have gained popularity owing to their efficiency [2]. Moreover, one can establish a natural connection between the point cloud features and the corresponding graph structure for analysis [11]. Although graph-based methods have achieved substantial success, they still suffer from low efficiency construction of graph models. How to construct a suitable graph always remains an open problem in graph application. Moreover, regular graph edges only connect two nodes and, thus, can only model pairwise relationships. However, surfaces in a point cloud usually contain more than two nodes. Point-to-point graph edges are unable to model such complex multilateral relationships. Thus, we are motivated to develop more general hypergraph models for point clouds.

Graph and Hypergraph Signal Processing on Point Clouds: Graph signal processing (GSP) is a recent geometric tool [12] that has shown great promises. Due to its ability in characterizing underlying structures of datasets, graph signal processing has been applied in point clouds. The authors of [11], for example, develop a fast resampling for point clouds based on graph frequency and GSP filters. Other processing tasks, such denoising [13] and compression [14], can also be implemented by GSP. To generalize GSP to capture high-dimensional multi-lateral interactions, a tensor-based framework of hypergraph signal processing (HGSP) was recently proposed in [5]. Highly suitable for characterizing the multilateral interaction among noise in a point cloud, HGSP can play a significant role in applications such as clustering, classification, compression, and denoising. The work of [4] further developed new approaches to hypergraph spectrum analysis and processing in gray-scale point clouds based on HGSP.
In this work, we extend existing studies and develop an efficient HGSP-based segmentation method for the gray-scale point clouds.

3 Preliminaries

In this section, we provide the background and introduce some preliminaries of tensor and hypergraph signal processing before discussing technical details.

3.1 Tensor

Tensor is a high-dimensional algebraic entity which can be viewed as a multi-dimensional array. The order of a tensor is the number of indices used to label its components [15]. Indeed, a scalar is a zeroth-order tensor; a vector is a first-order tensor; a 2D matrix is a second-order tensor; a 3D cubic array of entries with 3 indices is a third-order tensor, and an $M$-dimensional array is an $M$th-order tensor [16]. In this paper, we denote an $M$th-order tensor by $A \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$, whose entry in position $(i_1, i_2, \cdots, i_M)$ is $a_{i_1 \cdots i_M}$. The dimension (length) of the $k$th order is notated as $I_k$. Consider an $M$th-order tensor $A \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$ with entries $a_{i_1 \cdots i_M}$ and an $N$th-order tensor $B \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ with entries $b_{j_1 \cdots j_N}$. We denote their tensor product by $C = A \odot B$ for which the product tensor $C \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M \times J_1 \times J_2 \times \cdots \times J_N}$ is of $(M+N)$th-order, with entries $c_{i_1 \cdots i_M j_1 \cdots j_N} = a_{i_1 \cdots i_M} \cdot b_{j_1 \cdots j_N}$.

3.2 Hypergraph Signal Processing

Hypergraph signal processing (HGSP) is a tensor-based framework that can describe the multi-dimensional interactions between nodes [5]. Within the HGSP framework, a hypergraph with $N$ nodes and longest hyperedge connecting $M$ nodes, is represented by an $M$-order $N$-dimension representing tensor which can also be decomposed via orthogonal CP decomposition, i.e.,

$$A = (a_{i_1 i_2 \cdots i_M}) = \sum_{r=1}^{N} \lambda_r \cdot f_r \circ \cdots \circ f_r, \quad A \in \mathbb{R}^{N} \times \mathbb{R}^{N} \times \cdots \times \mathbb{R}^{N} \ (M \text{ times}) \quad (1)$$

Typically, this tensor entry $a_{i_1 i_2 \cdots i_M}$ indicates whether nodes $\{v_1, v_2, \cdots, v_M\}$ are connected in an adjacency relationship. In the orthogonal CP decomposition, $f_r$’s are orthogonal basis called spectrum components and $\lambda_r$ are frequency coefficients related to hypergraph frequency. All spectrum components $\{f_1, \cdots, f_N\}$ form the hypergraph spectral space. We name $(f_r, \lambda_r)$ the spectral pair of the given hypergraph.

Moreover, to capture the overall spectral information of the hypergraph, we define a supporting matrix

$$P_s = \frac{1}{\lambda_{\text{max}}} \begin{bmatrix} f_1 & \cdots & f_N \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_N \end{bmatrix} \begin{bmatrix} f_1^T \\ \vdots \\ f_N^T \end{bmatrix}. \quad (2)$$
Other details, such as hypergraph Fourier transform, HGSP filter design and hypergraph frequency analysis are introduced in [5].

4 Methods

In this section, we introduce the details of point cloud segmentation based on the hypergraph spectral clustering for gray-scale point clouds. There are three stages in the segmentation: 1) estimate the hypergraph spectral space, 2) order and select the key hypergraph spectrum, and 3) segment via clustering in the optimized hypergraph spectral space. In the first stage, we estimate the hypergraph spectrum directly from observed point clouds based on the hypergraph stationary process instead of decomposing the constructed hypergraph, since the representing tensor is memory-inefficient and its orthogonal-CP decomposition is time-consuming. We then estimate the distribution of hypergraph frequency coefficients according to smoothness, and order the spectrum based on the hypergraph frequency. Finally, we select the low frequency (smooth) spectral content and cluster in the optimized spectral space to segment the point clouds.

4.1 Estimation of Hypergraph Spectral Space

We first introduce to hypergraph spectrum estimation based on hypergraph stationary process.

Stationarity is an important property in analyzing observation of random variables. The authors of [17] defined a graph stationary process within the graph signal processing framework [12], which describes the stationary property of the graph shifting. Moreover, [18] proposed a method to estimate the spectral components of the graph under the assumption of stationarity for the observed dataset. For point cloud datasets, the three coordinates of each point can be interpreted as three observations of a node from 3 different viewpoints, which reflects the structural information embedded in the spectrum. Thus, we can estimate the hypergraph spectrum components based on hypergraph stationary processing. Let $\mathbb{E}()$ denote expectation and $(\cdot)^H$ denote conjugate transpose. The hypergraph stationary process is defined as follows.

Definition 1. (Weak-Sense Stationary Process) A stochastic signal $x \in \mathbb{R}^N$ is weak-sense stationary over hypergraph $P_s$ if and only if

$$\mathbb{E}[x] = \mathbb{E}[P_\tau x]$$

$$\mathbb{E}[(P_{\tau_1} x)((P^H)_{\tau_2} x)^H] = \mathbb{E}[(P_{\tau_1+\tau} x)((P^H)_{\tau_2-\tau} x)^H]$$

hold for every integer $\tau \geq 0$, where $P = \lambda_{max}P_s$ and $P_\tau = P^\tau$.

Eq. [3] requires stationary signals to have constant mean, similar to that in stochastic processes. Since the entry of $P$ in position $(i, j)$ is in the position $(j, i)$ of $P^H$, $P^H$ can be interpreted as propagation in an opposite direction of $P$. Then, Eq. [4] implies that the covariance of stationary process only depends on the difference between two steps, i.e., $\tau_1 + \tau_2$[4]. With the definition of hypergraph stationary process, we have the following property.
Theorem 1. A stochastic signal $x$ is WSS if and only if it has zero-mean and its covariance matrix has the same eigenvectors as the hypergraph spectrum basis, i.e.,

$$E[x] = 0$$

$$E[xx^H] = V \Sigma_x V^H,$$

where $V = [f_1, f_2, \ldots, f_N] \in \mathbb{R}^{N \times N}$ is the hypergraph spectrum.

The proof can be found in [4]. This property indicates that, we can estimate the hypergraph spectral components from the eigenspace of the covariance matrix of observations. Thus, a hypergraph-based spectrum estimation can be developed. Given a gray-scale point cloud $s = [X_1 \ X_2 \ X_3] \in \mathbb{R}^{N \times 3}$ where $X_i$ is the coordinates in $i$th angle, we can treat each $X_i$ as an observation of the points and normalize them to zero-mean. With the normalized observations, we can directly obtain the hypergraph spectrum based on their covariance matrix.

4.2 Estimation of the Spectrum Distribution

One important issue in spectral clustering is the ordering of spectral components in order to identify and remove some less critical and redundant information. Within the framework of HGSP, we rank the spectral components according to their (nonnegative) frequency coefficients in decreasing order to which indicates relatively low frequency to high frequency [5]. Now the problem rests with the estimation of the spectrum distribution. In practical applications, large-scale networks are often sparse, thereby making it meaningful to infer that most entries of the hypergraph representing tensor in typical datasets are zero [19]. In addition, signal smoothness is a widely-used assumption when estimating the underlying structure of graphs and hypergraphs [20]. Thus, a general estimation of hypergraph coefficients can be formulated as

$$\min_{\lambda} \text{Smooth}(s, \lambda, f_r) + \beta ||A||_T^2$$

s.t. $A = \sum_{r=1}^{N} \lambda_r \cdot f_r \circ \ldots \circ f_r, \quad A \in A$;

$$||A||_T = \sqrt{\sum_{i_1, i_2, \ldots, i_M = 1}^{N} \delta_{i_1 i_2 \ldots i_M}^2}.$$

Here $||A||_T^2$ is the tensor norm, and set $A$ includes prior information on the tensor types, e.g., adjacency or Laplacian. The smoothness function of signal $s$ can be designed flexibly for specific applications.

Instead of the exact calculation of frequency coefficients in Eq. (7), our 3D point cloud segmentation only requires a general idea on the distribution of frequency coefficients. Thus, we can simplify the problem as follows. First, we limit our the tensor order to $M = 3$, i.e., each hyperedge has length 3, since 3 nodes
are the required minimum to construct a surface. We then use the total variation based the supporting matrix, denoted by $\text{TV}(s) = ||s - (1/\lambda_{\text{max}})Ps||^2_2$, to describe the smoothness over the estimated hypergraph. In addition, the tensor norm equals $||A||_T = \lambda^T \lambda$ as proved in [4]. Let $\sigma = \frac{1}{\lambda_{\text{max}}} \lambda$. The formulation to estimate the spectrum distribution can be rewritten as

$$
\min_{\sigma} \alpha \sum_{i=1}^{3} ||X_i - PsX_i||^2_2 + \beta \sigma^T \sigma \quad (10)
$$

s.t. $\sigma = [\sigma_1 \ \sigma_2 \ \cdots \ \sigma_N]^T$; \hspace{1cm} (11)

$0 \leq \sigma_r \leq \sigma_{\text{max}} = 1$; \hspace{1cm} (12)

$\sum_{r=1}^{N} \sigma_r f_{r,i_1} f_{r,i_2} f_{r,i_3} \geq 0$, \hspace{1cm} $i_1, i_2, i_3 = 1, 2, \cdots, N$; \hspace{1cm} (13)

$Ps = [f_1 \ \cdots \ f_N]\text{diag}(\sigma)[f_1 \ \cdots \ f_N]^T$. \hspace{1cm} (14)

Note that the we usually neglect the constraint $\sigma_{\text{max}} = 1$ in practice to reduce computation complexity, since we only need to order the frequency coefficients for segmentation. Thus, the formulation is convex and can be readily solved by using well known numerical recipes. From the estimated $\sigma$, we order the spectral components in decreasing order of the frequency coefficients, which also indicates an increasing order of hypergraph frequency [5].

### 4.3 Segmentation based on Hypergraph Spectral Clustering

From the estimated spectrum components and frequency coefficients, we can directly propose a segmentation method based on spectral clustering. The detail steps are summarized as Algorithm 1.

Usually, we can define a threshold in Step 7. We will show more instructions of selecting the leading components, and discuss the efficiency of hypergraph spectral space in Section 5.2.

## 5 Experiments

In this section, we compare the performance of the proposed point cloud segmentation method with some more traditional methods to test the efficacy of hypergraph spectral clustering.

### 5.1 Comparison with $k$-means Clustering

Since the clustering parts of the hypergraph spectral clustering is based on the $k$-means clustering, we first compare the performance of the proposed segmentation method with the traditional clustering of $k$-means. To implement the $k$-means, we cluster over each row of point cloud coordinates. For the hypergraph spectral
Algorithm 1 Segmentation based Hypergraph Spectral Clustering

1: **Input**: Point cloud dataset \( s = [X_1, X_2, X_3] \in \mathbb{R}^{N \times 3} \) and the number of clusters \( k \).

2: Calculate the mean of each row in \( s \), i.e., \( \bar{s} = (X_1 + X_2 + X_3)/3 \);

3: Normalize the original point cloud data to zero-mean in each row, i.e., \( s' = [X_1 - \bar{s}, X_2 - \bar{s}, X_3 - \bar{s}] \);

4: Calculate the eigenvectors \( \{f_1, \cdots, f_N\} \) for \( R(s') = s'(s'^H) \);

5: Estimate the normalized frequency coefficients \( \sigma_r \)'s by solving Eq. (10)

6: Rank frequency components \( f_r \)'s based on their corresponding frequency coefficients in decreasing order, i.e., \( \{f_1, f_2, \cdots, f_N\} \) where \( \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_N \).

7: Find the first \( E \) leading spectral components \( f_r \) with larger \( \lambda_r \) and construct a spectrum matrix \( M \in \mathbb{R}^{N \times E} \) with columns as the leading spectrum components.

8: Cluster the rows of \( M \) into \( k \) clusters using \( k \)-means clustering.

9: Cluster node \( i \) into partition \( j \) if the \( i \)th row of \( M \) is assigned to \( j \)th cluster.

10: **Output**: \( k \) partitions of the point clouds.

clustering, we select the first \( E \) key spectrum components based the frequency coefficients, for which there is a clear step difference with the next (i.e., the \((E+1)\)-th) frequency coefficients. Typically, the first two or three bases sufficiently satisfy this criterion. We will present more properties in Section 5.2.

We test the algorithms over a dataset of animals [21, 22, 23] shown as Fig. 2. In the first column of Fig. 2, we segment the a point cloud of a cat into six parts. In Fig. 2(d), the hypergraph spectral clustering can segment the limbs, the tail and the head into different colors, while the \( k \)-means clustering fails. Similar conclusions hold when segmenting the wolf and the gorilla, shown in the last two columns of Fig. 2. Especially for the gorilla data, different limbs are clustered to show four different colors in Fig. 2(c) whereas the traditional \( k \)-means is unable to distinguish its two legs in Fig. 2(f). From the test results, we can see that the hypergraph spectral clustering exhibit better performance than the \( k \)-means clustering because hypergraph can capture the underlying structure of the point cloud and effectively identify the redundant information for removal based on the frequency coefficients.

5.2 Comparison with Graph-based Methods

Next, let us show the comparison between the hypergraph-based method and graph-based clustering methods. We shall also examine the analysis of different spectrum properties.

Graph-based spectral clustering has been successful in segmentation. In graph-based methods, a graph representation, i.e., Laplacian or adjacency matrix, is constructed to generate the spectral space [24]. For example, GSP can apply a Gaussian model [11] to encode the local geometry information through an adjacency matrix \( W \in \mathbb{R}^{N \times N} \). Let \( s_i \in \mathbb{R}^{1 \times 3} \) be the coordinates of the \( i \)th point.
The edge weight between points $i$ and $j$ is defined by

$$W_{ij} = \begin{cases} e^{-\frac{||s_i - s_j||^2_2}{2\delta^2}}, & ||s_i - s_j||^2_2 \leq t; \\ 0, & \text{otherwise}, \end{cases}$$

where the variance $\delta$ and the threshold $t$ are parameters to control the weights of edges. Another common model is the Laplacian matrix. Similar to that in the graph adjacency matrix, a threshold can be set to determine whether a link exist between two nodes according to their distance. The weight of the similarity matrix $S$ is set to one or zero to indicate the link status. Next, form a diagonal degree matrix $D$ whose entry is the degree of each node. A Laplacian matrix is defined by

$$L = D - S$$

(16)

where $D$ is a diagonal matrix with the degree of each node on the diagonal. The Laplacian matrix $L$ is used to facilitate the spectral clustering.

However, a limitation of graph-based model is that the 2D edges can only model pairwise relationship but is unable to capture the multilateral relationship of the nodes over surfaces. On the other hand, our proposed hypergraph model of point clouds overcomes this shortcoming and can extract the multilateral information. We will compare our hypergraph-based method with the aforementioned graph models to show the power of HGSP in processing 3D point clouds.

We compare the proposed hypergraph spectral clustering with the Laplacian-based method in Eq. (16) and the adjacency-based method in Eq. (15), respectively, over the same datasets of human and animals as described in Section 5.1. We shall first show some experimental results from the different point cloud
datasets given in Fig. 3. From the test results, we can see that the GSP method and HGSP method often have similar performance by clustering limbs and torso into different parts. However, the HGSP-based method is able to further distinguish tails and different legs clearly, whereas the GSP method sometimes fails to cluster the limbs clearly. We can conclude that the hypergraph model can capture the overall structural information of 3D point cloud better than a normal graph. The Laplacian-based method emphasizes the details in some complex structures. For example, in the gorilla results shown in Fig. 3(f), the Laplacian-based method can segment the feet from the legs and hands from the arms, respectively. Generally, both HGSP-based and GSP-based methods present clearer segmentation of the main features for the point cloud datasets. Laplacian-based methods can add some specific details in complex structures if there may be more clusters.

We are interested in reasons behind the performance difference of different graphical methods. To explore the causes that lead to the differences, we ex-
amine the distribution of eigenvalues or the frequency coefficients of different methods in the specific horse point cloud shown in Fig. 4. In different rounds of the experiment, we randomly sample 400, 1400, 2400 and 3400 points from the original horse point cloud and calculate the eigenvalues from different methods. The results are shown in Fig. 5, Fig. 6 and Fig. 7. The Y-axis is the normalized values of eigenvalues or frequency coefficients. The X-axis is the position of each eigenvalues, i.e., $\text{Pos}_i = i/N$ for the $i$th eigenvalue of $N$ nodes. From the results, we can see that HGSP-based method and GSP-based method have quite similar distributions, which indicate that their information is more concentrated to the first few key spectral components. Moreover, with increasing number of samples, the information becomes more concentrated among the key components. Compared to GSP-based method, the key information of hypergraph spectrum are more concentrated. That explains why HGSP-based and GSP-based methods have similar performances in the segmentation results of Fig. 3. Unlike the adjacency-based methods, the distribution of eigenvalues of the Laplacian shows a quite different curve in Fig. 7. This difference accounts for the performance difference between Laplacian-based segmentation from the GSP-based and HGSP-based methods based on adjacency. We also test segmentation over point cloud with different number of samples, with results given in Fig. 8. From the test results, we can see that the performance of segmentation naturally improves with increasing number of samples. For example, in Fig. 8(b), the blue nodes appear in two legs, while the Fig. 8(c) shows a clear segmentation of the two hind legs.
5.3 Analysis of the Time Complexity

We also test the computation running time of different methods over the animal datasets. From the test results summarized in Table 1, we can see that the k-means method is the fastest. It is because graph based methods need to estimate the spectrum before clustering. The GSP-based and Laplacian-based methods need more time, since they need to first calculate pair distances for every node to obtain the graph structure where we can directly estimate the HGSP spectral components. Note that, we do not need to calculate the exact values of frequency coefficients when implementing hypergraph spectral clustering. In fact, we only need an approximate distribution of the frequency coefficients to complete the segmentation task.

6 Conclusions

This work proposes a novel segmentation method for 3D point cloud based on hypergraph spectral clustering. We first estimate the hypergraph spectral space via hypergraph stationary processing, before ranking the spectral components according to their frequency coefficients. We further introduce a segmentation algorithm that utilizes the estimated hypergraph spectrum pairs. The test results over multiple point cloud datasets clearly demonstrate the effectiveness of hypergraph spectrum estimation and the performance advantage of the proposed hypergraph method for point cloud segmentation.

Owing to the power of HGSP in processing complex, high-dimensional structures, we anticipate many future successful applications of HGSP in 3D point clouds. Given robustly estimated spectrum components and frequency coeffi-
cients, many critical data processing tasks such as resampling, reconstruction, and feature extraction can benefit from hypergraph spectral analysis.
1. Rusu, R., Cousins, S.: 3d is here: point cloud library (pcl). In: 2011 IEEE International Conference on Robotics and Automation, IEEE (2011) 1–4
2. Nguyen, A., Le, B.: 3d point cloud segmentation: a survey. In: 2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM), IEEE (2013) 225–230
3. Cheng, J., Qiao, M., Bian, W., Tao, D.: 3d human posture segmentation by spectral clustering with surface normal constraint. Signal Processing 91(9) (2011) 2204–2212
4. Zhang, S., Cui, S., Ding, Z.: Hypergraph spectral analysis and processing in 3d point cloud. arXiv preprint arXiv:2001.02384 (2020)
5. Zhang, S., Ding, Z., Cui, S.: Introducing hypergraph signal processing: theoretical foundation and practical applications. IEEE Internet of Things Journal (2019)
6. Sappa, A.D., Devy, M.: Fast range image segmentation by an edge detection strategy. In: Proceedings Third International Conference on 3-D Digital Imaging and Modeling, IEEE (2001) 292–299
7. Besl, P.J., Jain, R.C.: Segmentation through variable-order surface fitting. IEEE Transactions on Pattern Analysis and Machine Intelligence 10(2) (1988) 167–192
8. Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M., Beetz, M.: Towards 3d point cloud based object maps for household environments. Robotics and Autonomous Systems 56(11) (2008) 927–941
9. Biosca, J.M., Lerma, J.L.: Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. ISPRS Journal of Photogrammetry and Remote Sensing 63(1) (2008) 84–98
10. Filin, S.: Surface clustering from airborne laser scanning data. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences 34(3/A) (2002) 119–124
11. Chen, S., Tian, D., Feng, C., Vetro, A., Kovačević, J.: Fast resampling of three-dimensional point clouds via graphs. IEEE Transactions on Signal Processing 66(3) (2017) 666–681
12. Ortega, A., Frossard, P., Kovačević, J., Moura, J.M.F., Vandergheynst, P.: Graph signal processing: Overview, challenges, and applications. Proceedings of the IEEE 106(5) (2018) 808–828
13. Schoenenberger, Y., Paratte, J., Vandergheynst, P.: Graph-based denoising for time-varying point clouds. In: 2015 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), IEEE (2015) 1–4
14. Thanou, D., Chou, P.A., Frossard, P.: Graph-based compression of dynamic 3d point cloud sequences. IEEE Transactions on Image Processing 25(4) (2016) 1765–1778
15. Zhang, S., Zhang, H., Li, H., Cui, S.: Tensor-based spectral analysis of cascading failures over multilayer complex systems. In: 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton), IEEE (2018) 997–1004
16. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Review 51(3) (2009) 455–500
17. Marques, A.G., Segarra, S., Leus, G., Ribeiro, A.: Stationary graph processes and spectral estimation. IEEE Transactions on Signal Processing 65(22) (2017) 5911–5926
18. Pasdeloup, B., Gripon, V., Mercier, G., Pastor, D., Rabbat, M.G.: Characterization and inference of graph diffusion processes from observations of stationary
signals. IEEE Transactions on Signal and Information Processing over Networks 4(3) (2017) 481–496

19. Segarra, S., Marques, A.G., Mateos, G., Ribeiro, A.: Network topology inference from spectral templates. IEEE Transactions on Signal and Information Processing over Networks 3(3) (2017) 467–483

20. Dong, X., Thanou, D., Frossard, P., Vandergheynst, P.: Learning laplacian matrix in smooth graph signal representations. IEEE Transactions on Signal Processing 64(23) (2016) 6160–6173

21. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Numerical geometry of non-rigid shapes. Springer Science & Business Media (2008)

22. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Efficient computation of isometry-invariant distances between surfaces. SIAM Journal on Scientific Computing 28(5) (2006) 1812–1836

23. Bronstein, A.M., Bronstein, M.M., Castellani, U., Falcidieno, B., Fusiello, A., Godil, A., Guibas, L.J., Kokkinos, I., Lian, Z., Ovsjanikov, M.: Shrec 2010: robust large-scale shape retrieval benchmark. Proc. 3DOR 5(4) (2010)

24. Ma, T., Wu, Z., Feng, L., Luo, P., Long, X.: Point cloud segmentation through spectral clustering. In: The 2nd International Conference on Information Science and Engineering, IEEE (2010) 1–4