3D-PIC: POWER ITERATION CLUSTERING FOR SEGMENTING THREE-DIMENSIONAL MODELS

Zahra Toony*, Denis Laurendeau*, Philippe Giguère**, Christian Gagné*

*Computer Vision and Systems Laboratory, Department of Electrical and Computer Engineering, Université Laval, Québec, QC, Canada
**Department of Computer Science and Software Engineering, Université Laval, Québec, QC, Canada

ABSTRACT

Segmenting a 3D model is an important challenge since this operation is relevant for many applications. Making the segmentation algorithm able to find relevant and meaningful geometric primitives automatically is a very important step in 3D image processing. In this paper, we adapted a 2D spectral segmentation method, Power Iteration Clustering (PIC), to the case of 3D models. This method is fast and easy to implement. A similarity matrix based on normals to vertices is defined and a modified version of PIC is implemented in order to segment a 3D model. The proposed method is validated on both free-form and CAD (Computer Aided Design) models, on real data captured by handheld 3D scanners, and in the presence of noise. Results demonstrate the efficiency and robustness of the method in all cases.

Index Terms — 3D mesh segmentation, 3D similarity matrix, meaningful clustering

1. INTRODUCTION

Segmenting a 3D model is an important step in different applications, such as object recognition and matching, mesh manipulation, texture mapping, reverse engineering and other applications which require the structure of the object shape. Segmenting a 3D model can be applied on point clouds [1, 2] or on meshes (with triangulation) [3, 4, 5, 6]. Two surveys presented in [7] and [8] on 3D mesh segmentation approaches classify the segmentation methods in different groups. One of the categories that attracted more attention is the spectral analysis group. The spectral graph theory was first introduced by [9] and then adapted by [10] as Normalized Cuts for image segmentation. These types of approaches define a similarity matrix of the model, calculate the Laplacian function based on similarities and determine eigenvectors in order to segment the models.

3D mesh segmentation using a spectral clustering method was done by [4] for the first time. It defines the similarity matrix based on likelihood of faces belonging to the same segment. This algorithm conducts the segmentation along concave regions and thus occasionally identifies unexpected segments. The authors then defined another 3D segmentation method in [5] based on partial similarity matrix. Afterwards, they combined the advantage of using spectral clustering to 2D contour analysis in [6]. This method needs some parameters to be set and is sensitive to noise. Recently, a method was introduced to segment 3D CAD mesh models [3]. This approach first classifies the CAD model into sparse and dense regions, and then uses the Hough transform and mean-shift segmentation to segment each region. However, it tends to fail in the presence of noise.

Power Iteration Clustering (PIC) was introduced in 2010 as a simple and scalable clustering algorithm [11]. The authors compared their method with spectral clustering approaches which differ in how a low-dimensional vector is found from the similarity matrix. In spectral clustering approaches, the bottom eigenvectors are segmented but in PIC, a weighted combination of eigenvectors is utilized and thus enabling the generation of results better or at least similar to those obtained using conventional spectral methods [11]. Here, we present a three-dimensional PIC (3D-PIC) algorithm, adapted from the 2D version.

The rest of the paper is organized as follows. The details of two-dimensional PIC are represented in Section 2. Section 3 presents our novel three-dimensional PIC. The experimental results and comparison with other methods is presented in Section 4 and finally, Section 5 concludes this paper.

2. 2D-PIC

The Power Iteration Clustering (PIC) method is a spectral analysis approach which first defines a similarity matrix and then embeds the data in a low-dimensional sub-space [11]. Spectral clustering algorithms, such as Normalized Cuts [10], usually find the Laplacian of the similarity matrix and perform the low-dimensional embedding on this matrix. However PIC finds a normalized similarity matrix instead of the Laplacian. It then performs the low-dimensional embedding with an approximation of the “eigenvector-weighted linear combination of all the eigenvectors of the normalized similarity matrix” [11]. This method is efficient in time and space, in comparison with other spectral analysis approaches and since it considers a weighted combination of all vertices, its results are also better or at least similar to those of other spectral clustering methods. The details of this method are presented in Algorithm 1.

3. OUR PROPOSED METHOD: 3D-PIC

As we desire to segment a 3D model in a meaningful way such as a human would do, we need to separate different meaningful geometric features from each other. In spectral analysis approaches, the features of the models are described through the similarity matrix. Therefore, a key component is a similarity measure that can maintain the primitives with the same properties in a segment.

To be able to discriminate different primitives like planes and cylinders in 3D space, and to find the edges between primitives we are proposing that the direction of the normals to the vertices...
Algorithm 1: The PIC algorithm [11]

Input: the similarity matrix $W$, diagonal matrix $D$, $(D_{i,i} = \sum_j W_{i,j})$ and number of clusters $k$.
Define $A$ as a row normalized similarity matrix:

$$A \leftarrow D^{-1}W.$$ 

Set $\epsilon = 10^{-5}$, ($V$: number of vertices).
Pick an initial vector $V^0$.
repeat
Set $V^{t+1} \leftarrow \frac{AV^t}{||AV||_1}$ and $\delta^{t+1} \leftarrow |V^{t+1} - V^t|$.
Increment $t$.
until $||\delta^{t+1} - \delta^t||_\infty \leq \epsilon$.
Use k-means to cluster points on $V^t$.
Output: Clusters $C_1, C_2, \ldots, C_k.$

can be a good measure. All normals of a plane are in the same directions and all normals of a cylinder are rotating around a circle (Figure 1b). Therefore, the cosine of the angles between vertices’ normals is a natural candidate for the weight function of the similarity matrix. In order to reduce the computational time, we identify the vertices that are connected by an edge and compute the cosine of the angle between their normals. The similarity between each vertex $i$ and $j$ is defined as:

$$W_{i,j} = \cos \langle \vec{n}_i, \vec{n}_j \rangle,$$

where, $\vec{n}_i$ and $\vec{n}_j$ are normal vectors at vertices $i$ and $j$, respectively.

Suppose the similarity matrix is calculated based on Equation 1. In this case, a color-coded representation of the edges’ values (similarities) is shown in Figure 1c, where red indicates the largest values and blue the lowest ones. Because of small differences between similarity values, the variations are not apparent enough and all values are shown in red. Hence, applying the Power Iteration Clustering method results in a meaningless segmentation outcome (Figure 1d). To make the method more reliable, we apply an exponential function on the weight values in order to decrease small weight values and increase large ones. This modification increases differences between values significantly and separates regions very well but, in order to assign a priority to the neighbours in comparison with the points that are far from each other, a small value $\eta$ is also added to the weights of neighbouring vertices. Therefore the similarity matrix is redefined as:

$$w(i, j) = \begin{cases} e^{\alpha \cos \langle \vec{n}_i, \vec{n}_j \rangle} + \eta & \text{if an edge connects } i \text{ and } j \\ 0 & \text{otherwise} \end{cases}.$$ \hspace{1cm} (2)

The color-coded representation of the new edge values (weights) is shown in Figure 1e. This new definition satisfies our goal of having large values of similarity where surface normals are similar (red), but small values at the boundaries between primitives (green). The result of applying the Power Iteration Clustering method on a 3D model using a redefined similarity matrix is presented in Figure 1f, which shows a well-segmented model. If we want to segment this 3D model into two clusters, at first glance, it consists of a cylinder and a cube-like shape which is similar to the result of our segmentation algorithm.

Algorithm 2 represents the steps of the 3D-PIC method proposed in this paper, which is an extension of the 2D-PIC algorithm introduced in [11]. This algorithm always converges, provided that a good initialization of $V^0$ is performed at step 4 [11]. The initialization value of $\epsilon$ is selected based on [11]. In step 7 of Algorithm 2, the $L_1$ norm is used instead of the $L_\infty$ norm, which leads to a slight increase in computation time for our algorithm but significantly improves the quality of the results. The values of constants $\alpha$ and $\eta$ are chosen empirically, but are kept the same across all of our experiments. The experimental results of 3D segmentation are presented in the next section.

4. EXPERIMENTAL RESULTS

Here we present the result of applying the proposed 3D-PIC method on different 3D models (both free-form and CAD models). The method is evaluated on real objects scanned with a handheld scan-

Algorithm 2: Proposed 3D-PIC method

Input: Provide the number of clusters, $k$, the constants $\alpha$ and $\eta$ to segment 3D model $G = (S, F)$, where $S$ is the set of vertices and $F$ is the set of faces.

1. Compute the similarity matrix $W$,

$$W_{i,j} = \begin{cases} e^{\alpha \cos \langle \vec{n}_i, \vec{n}_j \rangle} + \eta & \text{if an edge connects } i \text{ and } j \\ 0 & \text{otherwise} \end{cases}.$$ \hspace{1cm} (2)

2. Evaluate the diagonal matrix $D$, $D_{i,i} = \sum_j W_{i,j}$.
3. Compute the row normalized similarity matrix $A$, $A = D^{-1}W$.
4. Pick as initial vector $V^0$, $V^0(i) = \frac{\sum_j W_{i,j}}{\sum_j W_{j,j}}$, where, $V(W) = \sum_j W_{i,j}$.
5. Set $\epsilon = \frac{10^{-5}}{n}$, ($V$: number of vertices).
6. Set $V^{t+1} \leftarrow \frac{AV^t}{||AV||_1}$ and $\delta^{t+1} \leftarrow |V^{t+1} - V^t|$.
7. Increment $t$ and repeat step 6 until $||\delta^{t+1} - \delta^t||_1 \leq \epsilon$.
8. Use k-means, to cluster $V_t$ into $k$ segments.

Output: Clusters $C_1, C_2, \ldots, C_k$. 


ner and on 3D models corrupted by noise. The segmentation of such models reveals the robustness of our method even in the presence of noise.

All free-form 3D models used in the experiments are chosen from the Princeton Shape Benchmark for 3D Segmentation [12] and the 3D CAD models are taken from the database of Purdue University (ESB) [13]. The constants $\alpha$ and $\eta$ are set to 15 and 0.5, respectively. Figure 2 shows the result of our segmentation approach for an object with different numbers of clusters. This figure presents the effectiveness of our method in detecting various geometries in an object with an increasing number of clusters. In the figure, for each number of clusters, two sides of the 3D model are rendered in order to observe all sides of the object. As can be observed, by increasing the number of clusters our method detects more different geometric features on the object.

![Figure 2](image)

Figure 2: Segmenting a 3D model with different number of clusters. Column (a), (b), and (c) present the corresponding results for 5, 8, and 13 numbers of clusters, respectively.

The comparison of the results of the CAD model segmentation using our approach and a recently presented CAD model segmentation method [3] is presented in Figure 3. Xiao’s method segments the 3D CAD models automatically without selecting the number of clusters. The results of their method were obtained from the GUI that the authors gracefully made available to us. They use Hough transform, Gauss sphere mapping and meanshift to segment the 3D models. This method works well for sparse planes but for cylinders and cones, it sometimes fails in detecting or in associating parts together (which can be seen in Figure 3).

As presented in Figure 3, our method is capable of detecting planes and cylinders. For the first model, we present the segmentation results for $k = 2$, $k = 3$, and $k = 8$ clusters, which shows the effectiveness of our algorithm in identifying different primitives of a model.

The results of segmenting 3D objects scanned with a handheld scanner are also presented in Figure 4. The models were scanned with an accuracy of 2 mm with the Creaform Go!SCAN handheld 3D scanner. A noise removal method was applied on some of them (columns (a) and (d)) before segmentation [14].

To illustrate the robustness of our algorithm in the presence of noise, two models presented in Figure 3 were corrupted by a zero mean Gaussian noise with standard deviation $\sigma = 0.25$ and $\sigma = 0.4$ of mean edge length. Then, our segmentation approach was applied to the models. For each model, the results of the segmentation are shown in the case of noiseless and noisy models. Columns (c) and (d) of Figure 5 show a comparison between the approach presented in [3] and our method for noisy and noiseless models. The results show that our method segments the models almost similarly with or without noise, but reveal that Xiao’s approach is sensitive to noise.

Figure 5 provides more results of segmentation using our approach and demonstrates its ability to process both 3D free-form and CAD models.

5. CONCLUSIONS

In this paper, we presented a 3D segmentation method based on spectral analysis. This approach, named 3D-PIC, is an adaptation of the two-dimensional Power Iteration Clustering method. We defined an exponential weight function of vertex normals as the similarity measure. A vector of weighted eigenvectors is extracted from the similarity matrix using the PIC algorithm. We then applied $k$-means clustering on this vector to find the segments. Our segmentation approach is sensitive to noise. Then, our segmentation approach was applied to the models. For each model, the results of the segmentation are shown in the case of noiseless and noisy models. Then, our segmentation approach was applied to the models. For each model, the results of the segmentation are shown in the case of noiseless and noisy models.

![Figure 3](image)

Figure 3: Comparison with the method proposed by [3]. Column (a) shows the 3D model while column (b) and (c) present the results using our 3D-PIC method and the method proposed by [3], respectively.

![Figure 4](image)

Figure 4: Segmentation of real 3D models.
Figure 5: The result of our 3D segmentation approach in the presence of noise. Column (a) shows the 3D noisy and noiseless models. The normals of models presented in column (a) are shown in column (b). Column (c) and (d) are representing the results of segmentation by 3D-PIC and Xiao’s method [3], respectively.

Figure 6: More results provided by our 3D segmentation approach.

ACKNOWLEDGEMENTS

This work was supported by the NSERC-Creaform Industrial Research Chair on 3D Sensing. Z. Toony was supported by a FRQNT post-graduate scholarship. The authors thank Hongwei Lin for providing us with the GUI of their method [3].

6. REFERENCES


