Damanhour Journal of Intelligent Systems and Informatics
Volume 2 – Issue 1

Impact of Spectral Parameters on 3D Mesh Segmentation

Fatma Khairy[a,*], Mohamed H. Mousa[b], Hamed Nassar[a]

[a] *Computer Science Department, Suez Canal university, Ismailia, 41522, Egypt*
[b] *Computer Science and Artificial Intelligence Department, University of Jeddah,, Jeddah, 23890, Saudi Arabia*

Abstract

In this article, a thorough experimental investigation of three key spectral parameters is carried out to asses their impact on spectral segmentation. These parameters are the concavity threshold, c, the new segment size factor threshold, ν and the number of eigenvectors, K, of the affinity. For each parameter, experiments have been carried out to get an idea of the optimal value of the parameter, from the perspective of both the segmentation quality and the computational efficiency. The general conclusion of this work is that the optimal values of these parameters are model-dependent. The underlying causes for what properties in the model specify the optimal value of the parameter need deeply involved theoretical analysis, for which the present work has paved the way and provided intuitive pointers to consider.

Keywords: 3D mesh, spectral segmentation, eigenvector, Concavity Threshold.

1. Introduction

Three-dimensional mesh segmentation has emerged as a fundamental problem in computer graphics and geometric processing, with applications ranging from computer-aided design to biomedical imaging and virtual reality. As 3D scanning technologies and modeling tools become increasingly sophisticated, the need for robust and efficient mesh segmentation algorithms has grown correspondingly. Mesh segmentation refers to the process of partitioning a 3D polygonal mesh into meaningful sub-parts [1], either based on topological characteristics, geometric properties, or semantic considerations. Formally, given a 3D mesh $M = (V, E, F)$ consisting of vertices V, edges E, and faces F, the segmentation problem aims to find a partition $S = \{S_1, ..., S_k\}$

The challenge of 3D mesh segmentation lies in the inherent complexity of polygonal meshes, which may contain noise, non-uniform sampling, and various topological anomalies. Unlike image segmentation where pixels are arranged in a regular grid, mesh segmentation must account for irregular connectivity, varying vertex densities, and complex surface geometries.

Early approaches to mesh segmentation relied primarily on geometric criteria such as curvature, dihedral angles, or geodesic distances [2]. While these methods achieve reasonable success for simple objects, they often struggled with complex shapes, where local geometric features do not clearly correlate with semantic parts.

Traditional mesh segmentation algorithms can be broadly categorized into five classes. First, region-growing methods start from seed faces or

*Corresponding author
Email address: fatma_mahdi@ci.suez.edu.eg (Fatma Khairy)

vertices and iteratively expand regions based on similarity metrics. The expansion usually stops when encountering sharp geometric discontinuities or when certain size constraints are met [3].

Second, clustering-based approaches operate by grouping mesh elements in feature space using algorithms like k-means [4] or hierarchical clustering [5]. The key challenge here lies in defining appropriate distance metrics that capture both geometric and topological properties.

Third, Watershed algorithms adapted from image processing treat mesh curvature or other attributes as height fields, flooding basins from local minima until watershed boundaries emerge at ridges [6].

Fourth, primitive fitting methods attempt to approximate mesh regions with simple geometric primitives (planes, cylinders, etc.), with segmentation emerging from the fitting process [7].

Fifth, spectral segmentation methods, used in the present article, have gained prominence as they overcome many limitations of purely geometric approaches by incorporating global shape information through spectral graph theory [8]. These spectral techniques are rooted in the analysis of the mesh Laplacian matrix, which encodes both geometric and topological properties of the 3D mesh under consideration.

A key component of spectral segmentation is the Laplacian matrix L of a mesh, defined by

$$L = D - W \tag{1}$$

where W is the adjacency matrix encoding vertex connectivity (often with edge weights based on geometric properties), and D is the diagonal degree matrix. The eigenvectors of L capture intrinsic geometric information about the mesh, with the smallest non-zero eigenvalues corresponding to fundamental vibration modes of the shape. Computing the first K eigenvectors of the Laplacian, and projecting mesh vertices into the spectral embedding space, then performing clustering in this reduced-dimensional space, are typical activities in spectral segmentation.

The spectral approach offers several advantages, such as global consistency, as spectral methods consider the entire shape simultaneously, rather than relying on local decisions, and noise robustness, as the Laplacian's eigen structure tends to be stable under small perturbations.

In this paper, we build upon our previous work [9] by systematically evaluating the impact of the key spectral parameters on segmentation outcomes. Our goal is to highlight how specific choices in the spectral domain can significantly or insignificantly affect the accuracy, robustness, and interpretability of segmentation results. By conducting comprehensive experiments on established 3D mesh benchmarks [10], we provide insights and guidelines for selecting spectral parameters that enhance segmentation quality.

2. Related Work

Recent advances in spectral mesh segmentation have focused on improving computational efficiency, as well as combining spectral features with learning-based approaches [11]. The development of scale-invariant and anisotropic Laplacian operators has further enhanced the method's applicability to real-world meshes with non-uniform sampling and varying scales [12].

The authors of [13] present a spectral segmentation approach optimized for large-scale meshes. Their methodology employs edge collapse operations within a progressive mesh framework, developing a feature-preserving simplification technique that maintains the original mesh topology while optimally retaining geometric features during coarsening. Similarly, In [14] the authors propose a 3D segmentation technique based on Medial Axis Transform (MAT) analysis. Their method leverages the rich geometric and structural information contained in the MAT repre-

sentation to systematically detect and classify part junctions in complex 3D shapes through an efficient computational framework.

The authors of [15] propose spectral mesh segmentation via ℓ_0 gradient minimization, constructing a Laplacian matrix from local geometric/topological data and using its Fiedler vector to enforce segment uniformity through gradient optimization. Also, in [16] the authors present a feature-aware region fusion method, first over segmenting meshes via adaptive space partitioning, then iteratively merging regions using novel intra/inter-region difference metrics and fusion conditions based on shape features.

Comprehensive surveys on 3D mesh segmentation methods can be found in [17], [18], [19], [1], and [20], while [21] specifically focuses on the segmentation of 3D point cloud data.

Several other studies have focused on developing adaptive concavity thresholding techniques that adjust the threshold based on local image characteristics. For instance, the author of [22] present an improved approach for mesh segmentation that extracts concave and convex feature regions simultaneously. Similarly, the authors of [23] introduce a spectral framework where local geometry affinities are coupled with surface patch affinities, and the segmentation relies on processing each eigenvector of the heterogeneous graph Laplacian individually. The authors of [24] also highlighted the importance of concavity, developing an algorithm that favors segmentation along concave regions, which is inspired by human perception.

Several methods have been proposed for selecting the optimal number K of eigenvectors. The authors of [25] introduce a spectral clustering algorithm that employs only informative/relevant eigenvectors for determining the number of clusters and performing clustering, measuring the relevance of an eigenvector according to how well it can separate the data set into different clusters. More recently, The authors of [26] have proposed an approximate spectral clustering algorithm with eigenvector selection and self-tuned K, using two relevance metrics for estimating K. As highlighted by [27], spectral clustering is known to be particularly sensitive to its parameter choices, including the number of eigenvectors selected during the spectral embedding process.

The size of newly-generated segments directly affects the trade-off between over-segmentation and under-segmentation. A small segment size threshold may lead to over-segmentation, where objects are divided into many small segments. Conversely, a large segment size threshold may result in under-segmentation, where distinct objects are merged into a single segment. Therefore, selecting an appropriate segment size threshold is crucial for achieving optimal segmentation quality. The authors of [28] propose a new algorithm for image segmentation and demonstrate that the proposed algorithm is able to segment the objects properly regardless of their size.

3. Materials and Methods

In this section we provide a description of the underlying spectral segmentation theory, in order to get a feel for the experimental work which is the core of the present article.

Having converted a 3D mesh from face specification to patch specification, the starting point is to construct the affinity weight matrix W^P. Next, initial segmentation is used to create initial surfaces to be used in a heterogeneous graph, incorporating useful geometric relations. Lastly, spectral segmentation of the 3D mesh is carried out using the eigenvectors of W^P, described next.

The patch creation algorithm ends up forming set P of patches of M patches, henceforth referred to as nodes. In the sequel, patch i will be considered a node and represented by a positive integer p_i to identify it. We search for pairs of patch nodes which are neighbors. Then the

mesh can then be represented by an undirected graph $G = (P, E)$, where $P = \{p_i\}$ is a set of the M nodes and $E = \{\widehat{e}_{ij}\}$ is a set of the common edges between the neighbor nodes of the graph.

Define l_{ij} as the length of edge $\widehat{e}_{ij}$ and $\bar{l}$ as the average length of all edges $\widehat{e}_{ij}$ for all i and all j. Denote by $d_{i,j}$ the distance between these the two faces i and j. Then, we can find the affinity parameter

$$x_{ij} = \begin{cases} \frac{c^2}{d_{i,j}^2} & \text{if } d_{i,j} \geq c \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

where c is the concavity threshold. The value of c to result in good segmentation will be explored in this work.

With the above in mind, we will now construct, for the graph, G, the patch-to-patch affinity matrix

$$W^P = [w_{ij}^P], \qquad i = 1, 2, \ldots, M, \qquad j = 1, 2, \ldots, M, \tag{3}$$

where

$$w_{ij}^P = \begin{cases} \frac{l_{ij} x_{ij}}{\bar{l}} & \text{if } p_i \text{ and } p_j \text{ are neighbor} \\ 0 & \text{otherwise} \end{cases}.$$

is the weight of edge $\widehat{e}_{ij}$ and represents the affinity between nodes p_i and p_j.

A heterogeneous graph is one that contains different types of nodes and different types of edges. The initial segmentation is produced using the k-means clustering algorithm [4] to generate N regions, each a set of patches. Let $R = \{r_1, r_2, \ldots, r_N\}$ be the set of regions generated by the k-means algorithm, with each r_i a node referring to the set of patches it contains.

An illustration of the resulting heterogeneous graph is shown in Figure (1), where there are $M + N$ nodes of two types, those representing patches and those representing regions, and 3 types of arcs, those representing patch-to-patch, those representing patch-to-region, and those representing region-to-region.

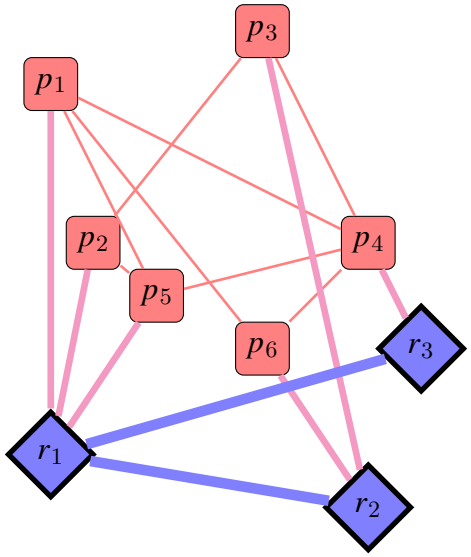


Figure 1. Heterogeneous graph model with rectangular patches and diamond-shaped regions. Three edge types visualize different connection strengths: thin (patch-patch), medium (patch-region), and thick (region-region).

The region-to-region affinity matrix W^R can be constructed (3) as follows.

$$W^R = [w_{mn}^R], \qquad m = 1, 2, \ldots, N, \qquad n = 1, 2, \ldots, N, \tag{4}$$

where

$$w_{mn}^R = \frac{\sum_{i \in r_m, j \in r_n} (w_{ij}^P)}{L},$$

with L being the number of edges common to the two neighbor regions r_m and r_n.

Using (3) and (4), we construct for the entire heterogeneous graph, the $(M+N) \times (M+N)$ heterogeneous weight matrix

$$W^H = \begin{bmatrix} W^P & A \\ A^T & W^R \end{bmatrix}, \tag{5}$$

where

$$A = [a_{ij}], \qquad i = 1, \ldots, M, \qquad j = 1, \ldots, N, \tag{6}$$

and a_{ij} are non-zero if patch i is in region j and is zero otherwise, i.e.

$$a_{ij} = \begin{cases} c & \text{if} \quad p_i \in r_j \\ 0 & \text{otherwise,} \end{cases}$$

with $c = 0.01$ found [23] to yield a reasonable concavity threshold.

This completes the construction of the heterogeneous graph, which forms the foundation for the improved spectral segmentation.

3.1. Spectral segmentation

Now, we compute K eigenvectors, arbitrarily labeled $U_1, U_2, \ldots, U_K$ and K corresponding eigenvalues, $\lambda_1, \lambda_2, \ldots, \lambda_K$ eigenvalues. The computation of the U_i and λ_i is attained by solving

$$L^H U_i = \lambda_i U_i \tag{7}$$

where the ij element of L^H is given by

$$l_{ij}^H = \begin{cases} \sum_{i=1}^{i=M+N} w_{ij}^H & \text{if } i = j \\ -w_{ij}^H & \text{otherwise} \end{cases} \tag{8}$$

being the elements of the unnormalized Laplacian L^H of the heterogeneous affinity matrix W^H. When the eigenvectors are obtained, only the first M elements of each eigenvector U_i are considered, corresponding to the M patches in the graph. It should be noted that each element U_{i_j} of U_i corresponds to a certain patch in the mesh.

The question remains open as to what value of K provides valid segmentation and at the same time does not deteriorate the computational efficiency (both time and space). This question is to be answered practically in the present paper, in particular in the next section.

Given a segment s_k, to divide it into two segments g_1 and g_2, using an eigenvector $U_i = (u_{i_1}, ui_2, \ldots, u_{i_M})$, we first identify the set Γ of the elements of U_i that correspond to the patches of segment s_k. Next we try to partition Γ into two subsets Γ_1 and Γ_2 obtained by

$$g_{1,2} = \underset{\Gamma_1,\Gamma_2}{\text{argmax}} \sum_{j,k} |U_{i_{j\in\Gamma_1}} - U_{i_{k\in\Gamma_2}}|\theta, \tag{9}$$

where $\theta = 0$ if the patches corresponding to j and k are not neighbors, $\theta = \exp(-w^P(j,k))$ if the patches corresponding to j and k are neighbors and the common edge between them is concave, and $\theta = 0.01$ otherwise. The value 0.01 has been found to give good segmentation results.

After partitioning a segment s_k into two segments g_1 and g_2, it should be ensured that the two segments meet these two segmentation conditions:

- Size condition: For a segment to be valid, its size, i.e. the number of patches it contains must be no less than a certain threshold, νM. That is, both $|g_1| > \nu M$ and $|g_2| > \nu M$ should hold.
- Connectedness condition: The patches in a segment must be all connected, i.e. neighbors to one another.

It is unclear in the existing literature what value for the size factor ν provides for good segmentation results, which has motivated us to explore the impact of this factor on segmentation quality.

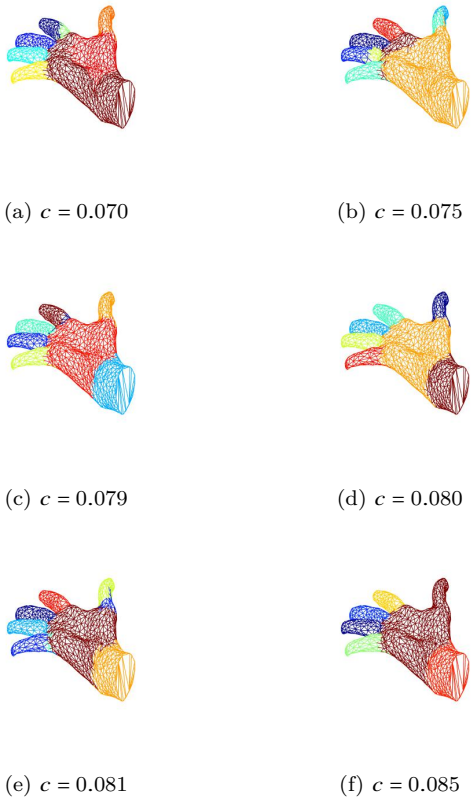
4. Experimental Work

In this section, the impact of three key spectral parameters on the segmentation will be investigated. These parameters are the concavity threshold, c, the size factor threshold, ν and the number of eigenvectors, K. For each parameter, experiments will be carried out to get an idea of the optimal values of these parameters from the perspective of both the segmentation quality and the computational efficiency. The main objective of this experimental investigation is lay the foundation for a meticulous theoretical analysis. All

the experiments of the present work have been carried out on a computer with an Intel Core i7 processor, 2.4 GHz, equipped with 8 giga bytes of RAM.

4.1. Concavity Threshold, c

To investigate the impact of the value of the concavity threshold, c, an arbitrary model will be chosen and the value of c will be changed, till an accurate segmentation evolves and beyond. The model chosen for this experiment is that of a hand. The aim of the experiment is to show that there is an optimal value of c, where values above and below result in segmentation errors. The results of this experiment are displayed in two figures. Figure (2) shows a visual assessment of the impact of different values of c on the segmentation of the hand model. As can be seen, the optimal value of c for this model is 0.08. Values above and below this value result in segmentation errors as can be seen. Besides the visual display of the impact of c, Figure (3) shows two quantified measures of the quality of segmentation for all the values of c used in this experiment, Rand index and cut discrepancy error. Both measures indicate that $c = 0.08$ provides the optimal segmentation.



(a) $c = 0.070$ (b) $c = 0.075$ (c) $c = 0.079$ (d) $c = 0.080$ (e) $c = 0.081$ (f) $c = 0.085$

Figure 2. Hand segmentation using different values of concavity threshold, c. The optimal is $c = 0.08$.

4.2. Newly-generated Segment Size Factor Threshold, ν

To investigate the impact of the value of the size factor threshold, ν, an arbitrary model will be chosen and the value of ν will be changed, till an accurate segmentation evolves and beyond. The model chosen for this experiment is that of a pair glasses. The aim of the experiment is to show that there is an optimal value of ν, where values above and below result in segmentation errors.

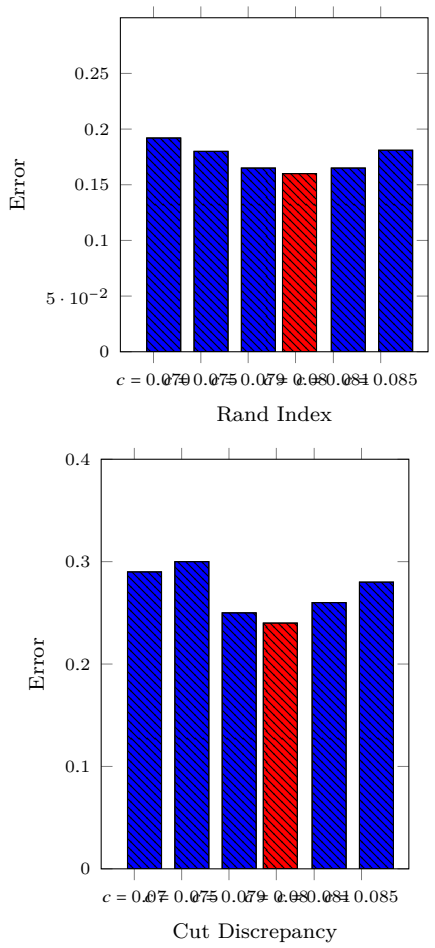
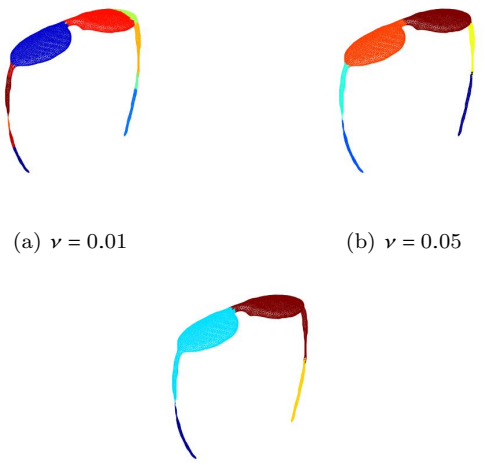


Figure 3. Rand index and cut discrepancy error for different values of the concavity threshold, c, used in the segmentation of the hand model.



(a) $\nu = 0.01$ (b) $\nu = 0.05$

(c) $\nu = 0.10$

Figure 4. Glasses segmentation using different values of size factor threshold, ν. The optimal is $\nu = 0.05$.

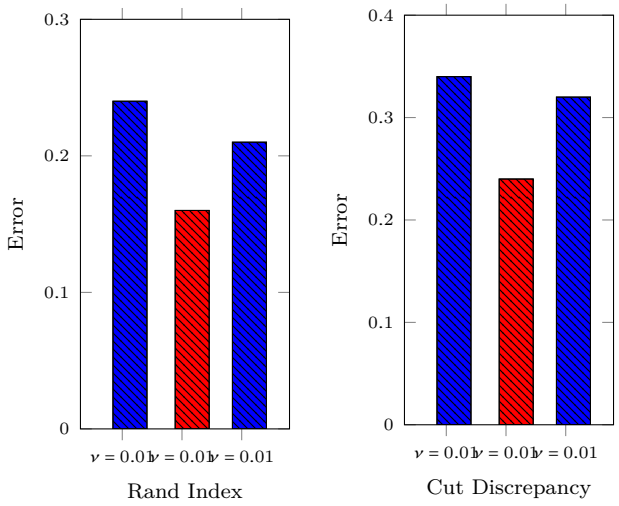


Figure 5. Rand index and cut discrepancy error for different values of the size factor threshold, ν, used in the segmentation of the glasses model.

The results of this experiment are displayed in two figures. Figure (4) shows a visual assessment of the impact of different values of ν on the segmentation of the glasses model. As can be seen, the optimal value of ν for this model is 0.05. Values above and below this value result in segmentation errors as can be seen. Besides the visual display of the impact of ν, Figure (5) shows two quantified measures of the quality of segmentation for all the values of ν used in this experiment, Rand index and cut discrepancy error. Both measures indicate that $\nu = 0.05$ provides the optimal segmentation.

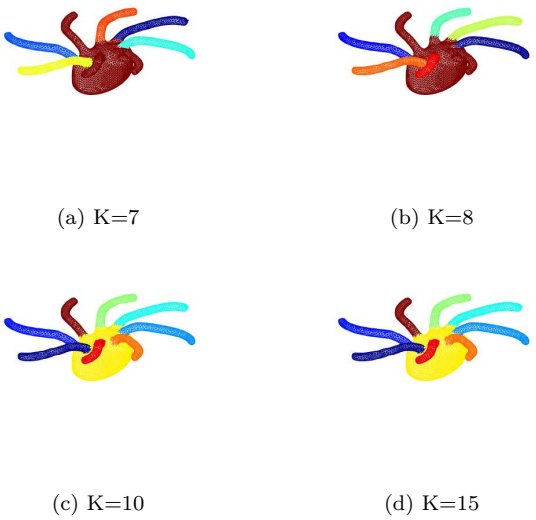
4.3. Number, K, of Eigenvectors

To investigate the impact of the value of the number of eigenvectors, K, three arbitrary models will be segmented and the value of K will be changed, till an accurate segmentation evolves. Unlike the above two parameters, c and ν, here once the accurate segmentation is obtained, no higher values of K are used. That is because the segmentation will remain accurate no matter how bigger K is made. However, the larger the

K the worse the computational efficiency (time and space).

The results of this experiment are displayed in three figures. Figure (6) shows a visual assessment of the impact of different values of K on the segmentation of an octopus model. As can be seen, the optimal value of K for this model is 10. As can be seen, the two values below this value result in segmentation errors. By contrast, the value above, $K = 15$, does not affect the segmentation quality, either positively or negatively, as can be seen, but of course degrade the computational efficiency. Figure (7) shows a visual assessment of the impact of different values of K on the segmentation of a hand model. As can be seen, the optimal value of K for this model is 5. As can be seen, the two values below this value result in segmentation errors. By contrast, the value above, $K = 10$, does not affect the segmentation quality, either positively or negatively, as can be seen, but of course degrade the computational efficiency. Figure (8) shows a visual assessment of the impact of different values of K on the segmentation of a chair model. As can be seen, the optimal value of K for this model is 4. As can be seen, the two values below this value result in segmentation errors. By contrast, the value above, $K = 10$, does not affect the segmentation quality, either positively or negatively, as can be seen, but of course degrade the computational efficiency.

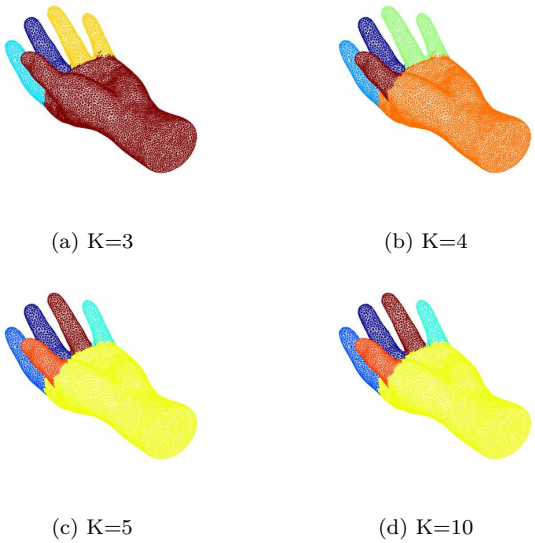
In conclusion, a large K parameter degrades the segmentation efficiency (space and time). However, a sufficiently small K can lead to an erroneous and invalid segmentation. Accordingly, it is important to identify the optimal value of K that guarantees valid segmentation and at the same time does not deteriorate the efficiency.



(a) K=7 (b) K=8

(c) K=10 (d) K=15

Figure 6. Number K of eigenvectors used to segment an octopus model. The optimal is $K = 10$.



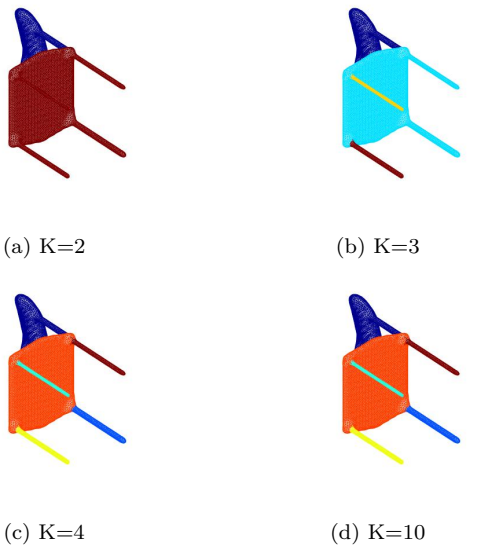
(a) K=3 (b) K=4

(c) K=5 (d) K=10

Figure 7. Number K of eigenvectors used to segment a hand model. The optimal is $K = 5$.

5. Conclusions

In this article, a thorough experimental investigation of three key spectral parameters is car-



(a) K=2 (b) K=3

(c) K=4 (d) K=10

Figure 8. Number K of eigenvectors used to segment a chair model. The optimal is $K = 4$.

ried out to asses their impact on spectral segmentation. These parameters are the concavity threshold, c, the size factor threshold, ν and the number of eigenvectors, K. For each parameter, experiments have been carried out to get an idea of the optimal value of the parameter, from the perspective of both the segmentation quality and the computational efficiency. The general conclusion is that the optimal value of these parameters are model-dependent. The underlying causes for what properties in the model specify the optimal value of the parameter need deeply involved theoretical analysis, which is intended to be the topic of future work. The present study has only paved the way for such analysis, providing intuitive pointers to focus on.

References

[1] A. Shamir, A survey on mesh segmentation techniques, Computer Graphics Forum 27 (6) (2008) 1539–1556.

[2] J.-M. Lien, N. M. Amato, Approximate convex decomposition of polyhedra and its applications, Comput. Aided Geom. Des. 25 (7) (2008) 503–522. doi:10.1016/j.cagd.2008.05.003.

[3] A. P. Mangan, R. T. Whitaker, Partitioning 3d surface meshes using watershed segmentation, IEEE Transactions on Visualization and Computer Graphics 5 (4) (1999) 308–321. doi:10.1109/2945.817348.
URL https://doi.org/10.1109/2945.817348

[4] U. Luxburg, A tutorial on spectral clustering, Statistics and Computing 17 (4) (2007) 395–416. doi:10.1007/s11222-007-9033-z.

[5] D. Khattab, H. M. Ebeid, A. S. Hussein, M. F. Tolba, 3d mesh segmentation based on unsupervised clustering, in: Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016 2, Springer, 2017, pp. 598–607.

[6] Y.-K. Lai, M. Jin, X. Xie, Y. He, J. Palacios, E. Zhang, S.-M. Hu, X. Gu, Metric-driven rosy field design and remeshing, IEEE Transactions on Visualization and Computer Graphics 16 (1) (2010) 95–108.

[7] A. Agathos, I. Pratikakis, S. Perantonis, N. Sapidis, P. Azariadis, 3d mesh segmentation methodologies for cad applications, Computer-Aided Design and Applications 4 (2007) 827–841.

[8] B. Lévy, H. R. Zhang, Spectral mesh processing, in: ACM SIGGRAPH ASIA 2009 Courses, SIGGRAPH ASIA '09, Association for Computing Machinery, New York, NY, USA, 2009, pp. 1–47. doi:10.1145/1665817.1665834.

[9] F. Khairy, M. H. Mousa, H. Nassar, Improving spectral segmentation of 3D meshes using face patches, Statistics, Optimization &

Information Computing (to appear in May 2025).

[10] X. Chen, A benchmark for 3d mesh segmentation, accessed Jan 1, 2025 (2025). URL https://segeval.cs.princeton.edu

[11] M. Masoumi, A. Hamza, Spectral shape classification: A deep learning approach, Journal of Visual Communication and Image Representation 43 (02 2017). doi:10.1016/j.jvcir.2017.01.001.

[12] J. Sun, M. Ovsjanikov, L. Guibas, Discrete laplace-beltrami operators for shape analysis and segmentation, Computers & Graphics 33 (3) (2009) 381–390.

[13] X. Bao, W. Tong, F. Chen, A spectral segmentation method for large meshes, Communications in Mathematics and Statistics 11 (3) (2023) 583–607.

[14] C. Lin, L. Liu, C. Li, L. Kobbelt, B. Wang, S. Xin, W. Wang, SEG-MAT: 3D Shape Segmentation using Medial axis Transform, IEEE Transactions on Visualization and Computer Graphics 28 (6) (2022, DOI:10.1109/TVCG.2020.3032566) 2430–2444. doi:10.1109/TVCG.2020.3032566.

[15] W. Tong, X. Yang, M. Pan, F. Chen, Spectral Mesh Segmentation via $\ell_0\ell 0$ Gradient Minimization, IEEE Transactions on Visualization and Computer Graphics 26 (4) (2020) 1807–1820. doi:10.1109/TVCG.2018.2882212.

[16] L. Wu, Y. Hou, J. Xu, Y. Zhao, Robust Mesh Segmentation Using Feature-Aware Region Fusion, Sensors 23 (1) (2023). doi:10.3390/s23010416.
URL https://www.mdpi.com/1424-8220/23/1/416

[17] P. Theologou, I. Pratikakis, T. Theoharis, A comprehensive overview of methodologies and performance evaluation frameworks in 3D mesh segmentation, Computer Vision and Image Understanding 135 (2015) 49–82. doi:https://doi.org/10.1016/j.cviu.2014.12.008.

[18] M. Rashad, M. Khamiss, M.-H. Mousa, A review on mesh segmentation techniques, International Journal of Engineering 6 (2017) 18–26.

[19] C. He, C. Wang, A survey on segmentation of 3d models, Wirel. Pers. Commun. 102 (4) (2018) 3835–3842. doi:10.1007/s11277-018-5414-1.

[20] R. Li, Q. Peng, 3d shape segmentation: A review, Recent Patents on Engineering 16 (5) (2022) 19–35.

[21] D. Krawczyk, R. Sitnik, Segmentation of 3D Point Cloud Data Representing Full Human Body Geometry: A Review, Pattern Recognition 139 (2023, DOI:10.1016/j.patcog.2023.109444) 109444. doi:https://doi.org/10.1016/j.patcog.2023.109444.

[22] M. Gu, L. Duan, M. Wang, Y. Bai, H. Shao, H. Wang, et al., An improved approach of mesh segmentation to extract feature regions, PLoS ONE 10 (10) (2015) e0139488. doi:10.1371/journal.pone.0139488.

[23] P. Theologou, I. Pratikakis, Theoharis, Unsupervised spectral mesh segmentation driven by heterogeneous graphs, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2) (2017) 397–410. doi:10.1109/TPAMI.2016.2544311.

[24] R. Liu, H. Zhang, Segmentation of 3d meshes through spectral clustering, in: Proceedings of the Computer Graphics and Applications, 12th Pacific Conference, PG '04,

IEEE Computer Society, USA, 2004, p. 298–305.

[25] T. Xiang, S. Gong, Spectral clustering with eigenvector selection, Pattern Recognition 41 (3) (2008) 1012–1029. doi:10.1016/j.patcog.2007.07.014.

[26] M. Alshammari, M. Takatsuka, Approximate spectral clustering with eigenvector selection and self-tuned k, Pattern Recogn. Lett. 122 (C) (2019) 31–37. doi:10.1016/j.patrec.2019.02.006.

[27] N. Analytics, A deep dive into spectral clustering methods for data analysis, accessed March 13, 2025 (2025).
URL `https://numberanalytics.com/blog`

[28] J. Li, H. Zhang, J. Wang, Y. Zhang, J. Li, Image segmentation based on constrained spectral variance difference and edge penalty, Remote Sensing 7 (5) (2015) 5980–6003. doi:10.3390/rs70505980.