Semi-global Quad Mesh Structure Simplification via Separatrix Operations

Kaoji Xu
University of Houston
cotrikxu@gmail.com

Muhammad Naeem Akram
University of Houston
makram2@uh.edu

Guoning Chen
University of Houston
gchen16@uh.edu

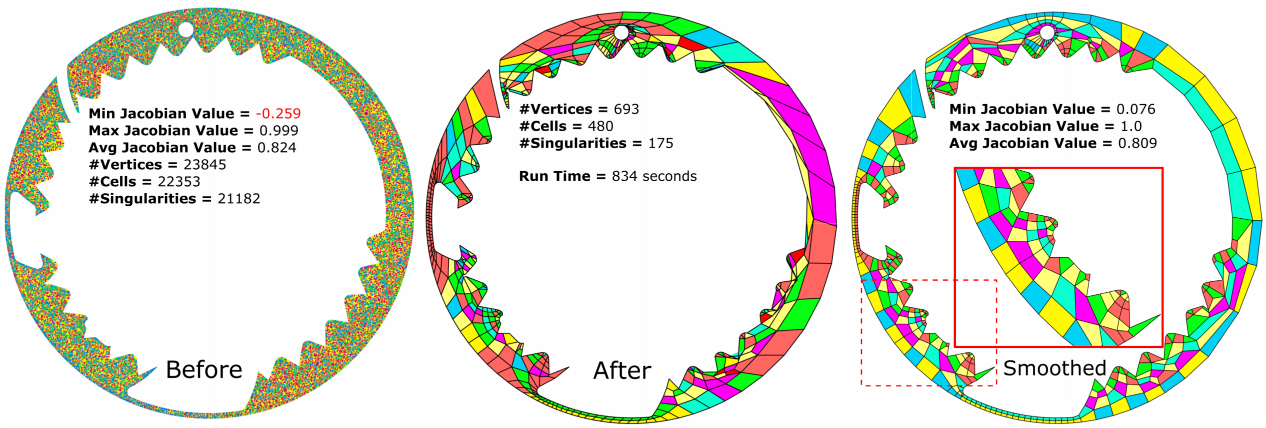


Figure 1: A result of our quad mesh structure simplification framework. Left: input; middle: simplified structure; right: optimized result. Different colored blocks in the right image show different base complex components. The statistics of the meshes before and after simplification are shown by the corresponding numeric values. This result shows that our framework not only can significantly reduce the structure complexity (99%), but also preserve boundary features.

ABSTRACT

This paper presents a semi-global method to simplify the structure of an all-quad mesh. The simplification aims to reduce the number of singularities, while preserving boundary features. The simplification operations of our method are based on the separatrices connecting adjacent singularities. The proposed semi-global method can handle quad-meshes with complex structures (e.g., quad-meshes obtained via Catmull-Clark subdivision of the triangle meshes) and produce quad meshes with much simpler structures.

KEYWORDS

quad mesh, structure simplification, separatrix-based

ACM Reference Format:

Kaoji Xu, Muhammad Naeem Akram, and Guoning Chen. 2020. Semi-global Quad Mesh Structure Simplification via Separatrix Operations. In *SIGGRAPH Asia 2020 Technical Communications (SA '20 Technical Communications), December 4–13, 2020, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3410700.3425436

SA '20 Technical Communications, December 4–13, 2020, Virtual Event, Republic of Korea

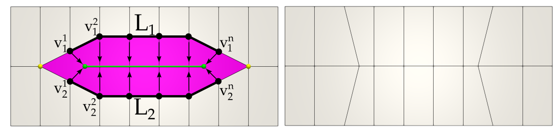
ACM ISBN 978-1-4503-8080-5/20/11…$15.00
https://doi.org/10.1145/3410700.3425436

1 INTRODUCTION

A quad mesh is a discrete manifold graph $G = \{V, E, F\}$ with a set of vertices V, edges E, and quad faces F. The valence of a vertex, Val_v is determined by the number of quads attaching to it. A *singularity* is an interior vertex with a valence other than 4 or a boundary vertex with a valence other than 2. It's index is defined as $1 - \frac{Val_v}{4}$. From each singularity with valence n we can trace out n curves, consisting of sets of edges that end at other singularities or the boundary of the mesh. We refer to these curves as the *separatrices*. All separatrices partition the quad mesh into a set of quadrilateral patches. This partitioning, leading to a coarse quad layout, is referred to as the *base complex* of the mesh (as shown in Figure 1).

Quad meshes are preferred in many medical and mechanical engineering applications due to their desired properties for numerical simulations. However, automatic generation of quad meshes with good structure (i.e., with as few singularities and as regular quad patches as possible) and high quality elements (e.g., large minimum scaled Jacobian) for arbitrary input remains a challenge despite numerous efforts [Bommes et al. 2013; Dong et al. 2006; Fang et al. 2018; Pietroni et al. 2016; Tarini et al. 2004].

Instead of generating a quad mesh from an arbitrarily complex input, an alternative is to simplify and optimize the structure of some initial meshes to achieve the ideal configuration. Two strategies have been proposed, i.e., the local approaches [Bozzo et al. 2010; Daniels et al. 2009; Tarini et al. 2010] and the global methods [II et al. 2008; Shepherd et al. 2010; Tarini et al. 2011]. Both strategies have



(a) before separatrix collapse (b) after separatrix collapse

(c) before separatrix split (d) after separatrix split

Figure 2: Simplification main idea : collapsing or splitting a local region (purple) formed by a 3–3 (green line) or 5–5 (yellow line) separatrix will remove 4 singularities (green and yellow dots).

been demonstrated effective for initial valid quad meshes with reasonably good structure. Nonetheless, local simplification cannot be guaranteed to generate an optimal structure or a near optimal one, while global simplification may not produce a boundary-conformal structure or may not be able to simplify the structure due to the constraint of boundary/feature preservation.

To address the limitations of the existing approaches, we introduce a semi-global simplification strategy for quad meshes with open boundaries that are often seen in mechanical engineering applications. Our strategy concentrates on the simplification of semi-global regions constructed based on the separatrices of the mesh. To remove degeneracy and achieve boundary conformality, we incorporate certain local connectivity modification operations introduced previously [Tarini et al. 2010]. The resulting framework is a comprehensive simplification framework centered at separatrices that can address complex mesh structure configurations, like those often seen in the quad meshes obtained from triangle meshes via Catmull-Clark subdivision, and significantly reduce them to much simpler ones (i.e., with fewer singularities and base complex components). See Figure 1 for an example result of our framework.

2 OUR METHOD

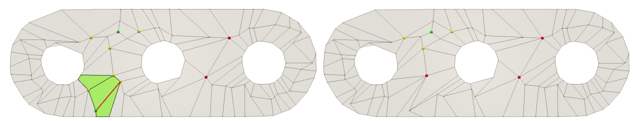
Our framework takes a valid all-quad mesh as input. First, it extracts all its singularities and their corresponding separatrices. Then, it performs the respective operations to reduce the number of singularities procedurally. The operations involved include the semi-global operations based on separatrices (Section 2.1) and additional local operations to prevent the potential degeneracy.

2.1 Operations based on Separatrices

In this work, we focus on the following four representative separatrix types, i.e., 3–3, 5–5, 3–5, and half separatrices, respectively.

Separatrix collapsing for 3–3 connection. Consider the configuration shown in Figure 2(a), there are three parallel lines (including the separatrix). All three lines have the same number of vertices as the separatrix (i.e., the middle line, denoted by $L_s = \{v_s^i\}$, where $i \in \{1, 2, ..., n\}$ and n is the number of vertices), and the two ending singularities are both valence-3, while the other vertices are regular. Assume $L_1 = \{v_1^i\}$ and $L_2 = \{v_2^i\}$ are the other two lines, $\{\text{val}_1^i\}$ and $\{\text{val}_2^i\}$ are their valences, $\{\text{feature}_1^i\}$ and $\{\text{feature}_2^i\}$ are their features (sharp edge or corner). To preserve the boundary feature, $\{v_1^i, v_2^i\}$ and v_s^i need to collapse to one of these 3 vertices. After collapsing, the four involved singularities are canceled.

Separatrix splitting for 5–5 connection. Instead of collapsing, we perform a splitting in the region surrounding a 5–5 connection. In Figure 2(c), a local region (purple) surrounding the separatrix (the yellow line) contains a pair of valence-5 singularities (yellow dots) and two valence-3 singularities (green dots). The splitting operation inserts two lines (dark lines in Figure 2(d)) parallel to the original separatrix, which removes the four singularities involved, i.e., they become regular. If any of the two end vertices in a region shown in Figure 2(c) is valence-5, splitting this region will produce a valence-6 singularity. A valence-6 singularity is acceptable, since it can be split into two valence-5 singularities.

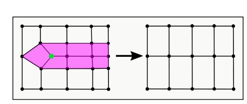


(a) before chord collapse (b) after chord collapse

Figure 3: Collapsing a chord enclosing a $3 - 5$ separatrix (a) results in a simplified quad-mesh (b). Different colored dots denote singularities with different valences.

Chord collapsing for 3–5 connection. To handle this configuration, a simple chord collapsing can be performed (Figure 3). The chord is formed by the edges that are parallel to one edge of the 3–5 separatrix. The chord collapsing will cancel at least 2 singularities (i.e., a valence-3 singularity collapses to a valence-5 one, producing a regular vertex). The details on the chord collapsing can be found in [II et al. 2008]. It is important to note that, not all chords can be collapsed, such as the one whose collapsing produces a singularity with a valence that is not in the user-defined range $[Val_{min}, Val_{max}]$ of the vertex valence (e.g., $Val_{min} = 3$ and $Val_{max} = 5$). Besides, a chord collapsing should not be performed if it changes the boundary (e.g., violating the boundary conformality requirement – all boundary vertices should have the ideal valences based on the angles around them).

Half-separatrix collapsing. For a quad-mesh with boundaries, a separatrix starting from an interior singularity (green dot) may end at a regular vertex on the boundary (see the inset to the right). We refer to this separatrix as a *half-separatrix*. Half-separatrix can be collapsed as illustrated by the inset, which is similar to the collapsing illustrated in Figure 2(a)(b).



What we described so far is an ideal semi-global region. However, in practice we may not always have such an ideal region, i.e., not having the two ending vertices with the desired valences as shown in Figure 2(a) or Figure 2(c). To enable further simplification of the

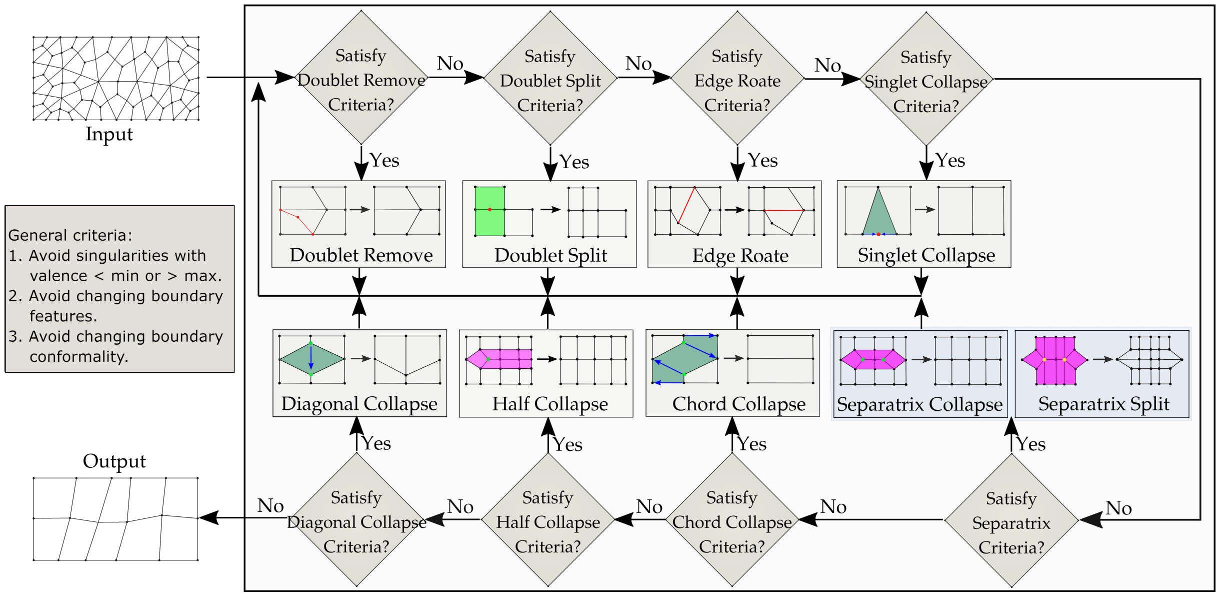


Figure 4: Our simplification pipeline.

mesh, we can relax the constraint (i.e., all vertex valences fall in the range $[Val_{min}, Val_{max})$ and allow valence greater than 3 for collapsing, less than 4 for splitting, or 3–n (n>5) for chord collapsing.

The above separatrix operations may result in degenerate configurations (e.g., doublets — valence 2 interior vertices) or may not achieve boundary preservation. To address this, we also incorporate the local operations introduced in the previous works [Bozzo et al. 2010; Daniels et al. 2009; Docampo-Sanchez and Haimes 2019; Tarini et al. 2010], including edge rotation, doublet removal, singlet collapsing, and diagonal collapsing.

Boundary Feature Preservation. Before simplification, we identify the feature vertices (regular, sharp edge, or corner) and extract the boundary feature lines. Vertices at the corners (i.e., boundary singularities or with angles outside a range, $[180^\circ - \theta, 180^\circ + \theta]$, θ is set by the user) are fixed. Vertices on sharp edges can only move along the sharp edges. Operations that alter a corner or a sharp edge are prevented to achieve a boundary configuration as close to the input boundary as possible.

2.2 The Complete Simplification Pipeline

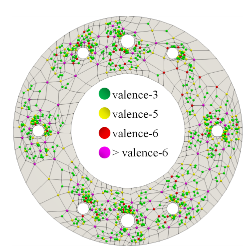
We illustrate our entire framework in Figure 4. The steps include: 1) For 3–3 separatrix, we perform separatrix collapsing; 2) For 5–5 separatrix, we perform separatrix splitting; 3) For 3–5 separatrix, we perform chord collapsing; 4) We perform edge rotation to achieve boundary conformality; 5) We perform doublet removal, doublet splitting, and singlet collapsing to remove singularities with unwanted valences; 6) We perform half-separatrix collapsing and diagonal collapsing to address the limitations of the above operations to continue canceling singularities. During the above steps, we preserve the singularity valence and surface features.

After the above structure simplification, we perform a feature preserved smoothing and resampling, adapted from a previous work [Xu et al. 2018]. During this optimization, we fix vertices located at the corner-like sharp features.

3 RESULTS

We applied our simplification method to a number of quad meshes with different geometry and topology configurations. Figure 5 shows the results of our simplification framework applied to a number of representative quad meshes. In particular, we achieve over **97%** reduction of the singularities for these models (Table 1), and the structures in the simplified meshes are close to their ideal structures given their boundary configurations.

Comparison with non-separatrix-based simplification. Usually, local operations can be applied to a quad-mesh that is split from a triangle mesh, while the global operations, such as chord collapsing, cannot because they may produce singularities with valences not falling in the range $[Val_{min}, Val_{max}]$. Singularity alignment is usually applied to a closed surface, while our method targets an open surface with boundary. Therefore, it is not reasonable to compare our results to the simplification obtained using singularity alignment. The existing techniques, no matter whether they perform local operations or chord collapsing, are usually applied to smooth surface with fewer features. That said, they may not be concerned with feature preservation. Nonetheless, as an initial comparison, we approximate the existing local methods by disabling the separatrix-based operations and applying to the 8-hole model. The result is shown in the inset. As can be seen, the obtained structure is not as



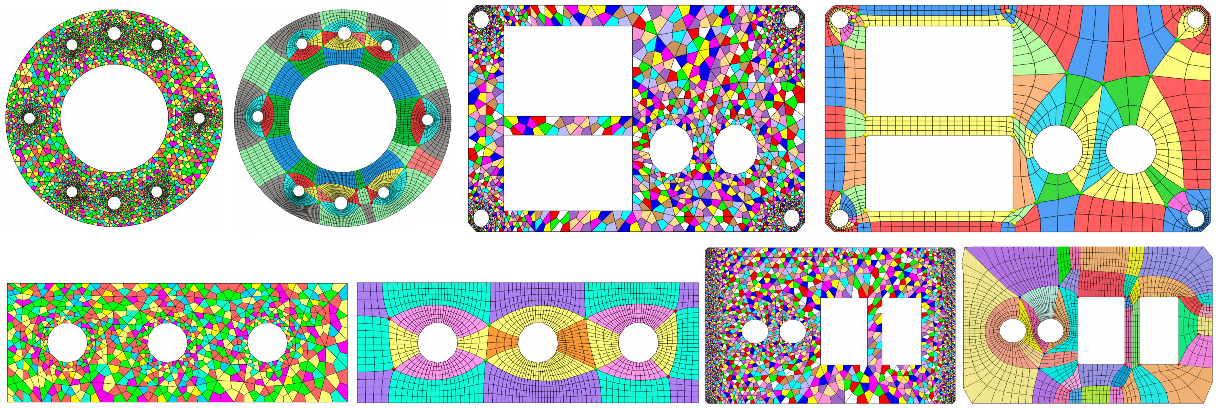


Figure 5: Simplification results of our framework on a number of representative quad meshes. Left image of each pair shows the input mesh while the right image shows the simplification result. Different colored blocks correspond to different base complex components.

simple as the one shown in Figure 5. In the future, more complete and rigorous comparison will be needed.

Table 1: Performance for our simplification method. #V, #F, and #S denote the number of vertices, faces, and singularities, respectively. R indicates the reduction ratio of the singularities, computed as $\frac{Input\#S - Output\#S}{Input\#S}$.

Model	Input			Output				
	#V	#F	#S	#V	#F	#S	**R**	Time (s)
8 holes	4234	4032	2109	335	229	18	**99.15%**	19.10
3 holes with corners	1056	975	523	144	96	12	**97.7%**	1.46
2 holes 2 squares	3671	3465	1825	140	97	20	**98.9%**	15.46
6 holes 2 squares	2253	2028	1112	352	242	23	**97.93%**	8.78
teaser	23845	22353	21182	693	480	175	**99.18%**	834.18

Performance. We have applied our simplification method to a number of models. Table 1 provides the timing information of our framework. All timing information was obtained on a workstation with Intel(R) Xeon(R) CPU E5-1620 v2 @3.70 GHz and 48 GB Memory @1866 MHz.

4 SUMMARY AND FUTURE WORK

We proposed an effective quad-mesh simplification method that is mainly based on a number of semi-global separatrix operations. Our method is heuristic and case-driven. The proposed semi-global operations aim at solving the limitations of the existing local and global operations. Combined with some local and global operations, our new framework can simplify a quad-mesh split from a triangle mesh to a mesh with much simpler structure compared to the existing methods.

Limitation. Despite the promising results obtained, there are still a few places to improve. First, we employ a naive ordering strategy for the separatrix simplification operations (i.e., based on their ids), which may lead to a sub-optimal structure that cannot be further simplified. Second, our algorithm may not place the resultant singularities to their optimal locations, which may be addressed by using certain guidance field derived from boundaries. Third, our current framework concentrates mainly on 2D patches with open boundaries, even though we can partition a surface into patches with sharp features as their boundaries. In the future, we plan to address the above limitations and extend our simplification pipeline for quad meshes to hex-meshes.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable feedback. This research was partially supported by NSF IIS 1553329.

REFERENCES

David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, and Denis Zorin. 2013. Quad-mesh generation and processing: A survey. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 51–76.

Agostino Bozzo, Daniele Panozzo, Enrico Puppo, Nico Pietroni, and Luigi Rocca. 2010. Adaptive Quad Mesh Simplification.. In *Eurographics Italian Chapter Conference*. Citeseer, 95–102.

Joel Daniels, Cláudio T. Silva, and Elaine Cohen. 2009. Localized Quadrilateral Coarsening. *Comput. Graph. Forum* 28, 5 (2009), 1437–1444.

Julia Docampo-Sanchez and Robert Haimes. 2019. Towards fully regular quad mesh generation. In *AIAA Scitech 2019 Forum*. 1988.

Shen Dong, Peer-Timo Bremer, Michael Garland, Valerio Pascucci, and John C. Hart. 2006. Spectral surface quadrangulation. *ACM Trans. Graph.* 25, 3 (2006), 1057–1066.

Xianzhong Fang, Hujun Bao, Yiying Tong, Mathieu Desbrun, and Jin Huang. 2018. Quadrangulation through morse-parameterization hybridization. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 92.

Joel Daniels II, Cláudio T. Silva, Jason Shepherd, and Elaine Cohen. 2008. Quadrilateral mesh simplification. *ACM Trans. Graph.* 27, 5 (2008), 148.

Nico Pietroni, Enrico Puppo, Giorgio Marcias, Roberto Scopigno, and Paolo Cignoni. 2016. Tracing field-coherent quad layouts. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 485–496.

Jason F. Shepherd, Mark W. Dewey, Adam C. Woodbury, Steven E. Benzley, Matthew L. Staten, and Steven J. Owen. 2010. Adaptive mesh coarsening for quadrilateral and hexahedral meshes. *Finite Elements in Analysis and Design* 46, 1 (2010), 17 – 32. https://doi.org/10.1016/j.finel.2009.06.024 Mesh Generation - Applications and Adaptation.

Marco Tarini, Kai Hormann, Paolo Cignoni, and Claudio Montani. 2004. PolyCube-Maps. *ACM Trans. Graph.* 23, 3 (2004), 853–860.

Marco Tarini, Nico Pietroni, Paolo Cignoni, Daniele Panozzo, and Enrico Puppo. 2010. Practical quad mesh simplification. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 407–418.

Marco Tarini, Enrico Puppo, Daniele Panozzo, Nico Pietroni, and Paolo Cignoni. 2011. Simple quad domains for field aligned mesh parametrization. *ACM Trans. Graph.* 30, 6 (2011), 142.

Kaoji Xu, Xifeng Gao, and Guoning Chen. 2018. Hexahedral mesh quality improvement via edge-angle optimization. *Computers & Graphics* 70 (2018), 17–27.