αMST: A Robust Unified Algorithm for Quadrilateral Mesh Adaptation in 2D and 3D

Chaman Singh Verma[a,*], Krishnan Suresh[b]

[a]*Department of Computer Sciences, University of Wisconsin, Madison, WI 53706, USA*
[b]*Department of Mechanical Engineering, University of Wisconsin, Madison, WI 53706, USA*

Abstract

Mesh adaptation plays a critical role in balancing computational efficiency and numerical accuracy. Three types of mesh adaptation techniques exist today, namely, mesh improvement, mesh refinement and mesh simplification, and, for each of these, several algorithms have been proposed. Current mesh adaptation algorithms yield acceptable geometric mesh quality, but provide limited control over topological quality.

In this paper, we introduce a unified algorithm for all three types of mesh adaptation, specifically for quadrilateral meshes. The algorithm builds upon the *Minimum Singularity Templates* (MST) proposed by the authors for improving the topological quality of a quadrilateral mesh. The MST is extended here to define the concept of an αMST where a single parameter α controls mesh adaptation: $\alpha = 1$ for mesh improvement, $\alpha > 1$ for mesh refinement, and $\alpha < 1$ for mesh simplification. The proposed algorithm generates a mesh that is adapted to user requirements of high geometric and topological qualities. Further, it is non-hierarchical and stateless, and yet it provides an arbitrary level of mesh adaptation. Finally, since cyclic chords can play an important role in quadrilateral mesh adaptation, we provide a simple constructive algorithm to insert such chords using αMST.

The proposed αMST templates can also be used to improve surface quadrilateral meshes using conformal mapping of surface charts. Furthermore, rotating the templates in the direction of cross-fields allows mesh edges to align

*Corresponding author. Tel.: +0-608-698-4729

along curvature lines. The proposed 3D generalization is fast, scalable, and inexpensive while improving alignment, shape, size, and sparsely placing the singularities. Several examples (both in 2D/3D) are presented that demonstrate the robustness, efficiency, and versatility of the proposed concept and algorithm.

Keywords: Quadrilateral mesh generation, Singularities, Quad templates.

1. Introduction

Since inception, automatic mesh generating algorithms have been continuously evolving to meet engineering (for example, see the surveys by Bommes [3] and Owen [25]). These algorithms typically accept user's requirements at a high level of abstraction, and produce a mesh with high geometric fidelity for simulation. These mesh generators have greatly simplified finite element simulations. The complete automation provided by these methods significantly reduce the most time-consuming phase of simulation, i.e. preparing a model from the underlying geometry. However, many simulation problems are dynamic in nature, i.e., to balance computational efficiency and numerical accuracy, even a high-quality mesh must be adapted. For example, in hypersonic flow simulation, the mesh must be refined near shock-waves, while in structural analysis, meshes must similarly be refined, and the quality improved near stress raisers; these critical regions are typically not known *a priori*. A naive and inefficient strategy would be to refine and improve the quality of the mesh everywhere, but this is impractical. It will lead to finite element models with large degrees of freedom, slowing down the simulation. Instead, meshes must be refined and improved in critical regions and coarsened elsewhere, a process called *mesh adaptation*. Mesh adaptation ensures a balance between computational efficiency and numerical accuracy.

Similar to mesh generation, for mesh adaptation to be effective, it must be fully automated, efficient, and versatile. Several such adaptation strategies have been proposed for both simplicial (triangular and tetrahedral) and non-simplicial (quadrilateral and hexahedral) meshes; the latter being significantly more challenging [1]. The focus of this paper is on quadrilateral mesh adaptation.

Once one or more regions have been identified within a mesh for adaptation, the overall goal is to improve, refine, or coarsen the mesh in these regions while respecting both geometric and topological quality constraints. Geometric quality constraints include aspect ratio, skew, distortion, shear, etc.; current adaptation strategies are typically capable of respecting such geometric constraints. The topological quality, on the other hand, is determined by the number of nodal *singularities* in the mesh; for a quadrilateral mesh, a mesh node is *regular* if it has four incident edges, otherwise it is *singular* (or irregular). Existing mesh adaptation strategies provide limited control over topological quality since it is considered hard to optimize and manipulate topology of a quad mesh, resulting in a large number of singularities. Excessive singularities can, unfortunately lead to (1) numerical instability in CFD applications [34], (2) wrinkles in subdivision surfaces [16], (3) irrecoverable element inversions near concave boundaries, (4) helical patterns [2], (5) produce visible seams in texture maps, and (6) breakdown of structured patterns on manifolds.

A second limitation of current mesh adaptation strategies is that they are specific to the type of mesh adaptation, i.e., different strategies are needed for mesh improvement, mesh refinement, and mesh simplification, and several such strategies must be combined in practice.

In this paper, we describe a unified and robust algorithm for quadrilateral mesh adaptation, with control over both geometric and topological qualities. The algorithm is based on the *Minimum Singularity Templates* (MST) proposed in [36]. While the MST was used to remove singularities in a mesh, it is extended here to define the concept of αMST where a single parameter α controls mesh adaptation: $\alpha = 1$ for mesh improvement, $\alpha > 1$ for mesh refinement, and $\alpha < 1$ for mesh simplification. A second salient feature of the proposed algorithm is that it is non-hierarchical and stateless, making it easy to implement.

The proposed method can be easily extended to 3D surfaces using classical parameterization techniques. This can be further improved if mesh edges are aligned in the direction of curvature. We do this by rotating the templates in the direction of the curvature. By aligning a large number of quadrilateral elements in each patch in the dominating direction, we gain a significant improvement in the algorithm efficiency. We demonstrate that in many cases our local approach produces result similar to some of the global methods which are usually expensive.

2. Basic Definitions and Proposition

In this paper, we use standard meshing terminology. However, for clarity of exposition, we reiterate few of them.

Definition 1. *The valence of a vertex v_i is the number of edges incident on it. A vertex with "n" valence is denoted by $\mathcal{V}n$. An internal vertex with valence 4 is considered regular, otherwise it is an irregular or a singular vertex. An internal vertex with valence 2 is called a doublet.*

In this paper, we consider only $\mathcal{V}3$ and $\mathcal{V}5$ singular nodes as all other high valence nodes can be converted into $\mathcal{V}3$ and $\mathcal{V}5$ nodes using standard atomic **face open** *or* **face close** *operation [1].*

Definition 2. *A patch is a sub-mesh with disc topology (Figure 1). Furthermore, we assume that the boundary nodes of the patch are ordered counter-clockwise. We designate a set of boundary nodes $N \in \{3, 4, 5, 6\}$ of the patch as corner nodes, and call the patch as N-sided patch. A* side *of the patch is defined as the mesh boundary between two consecutive corner nodes (Figure 2).*

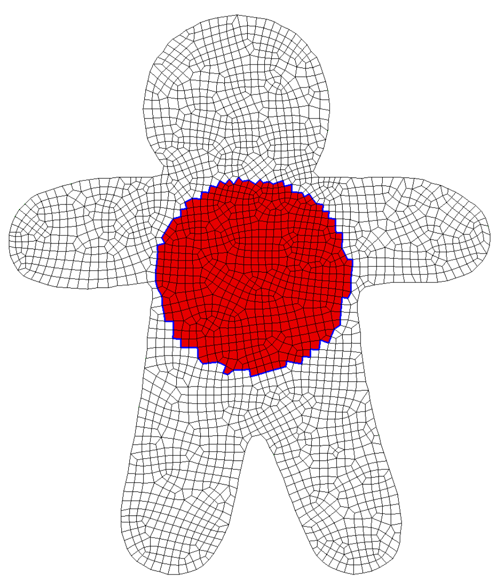
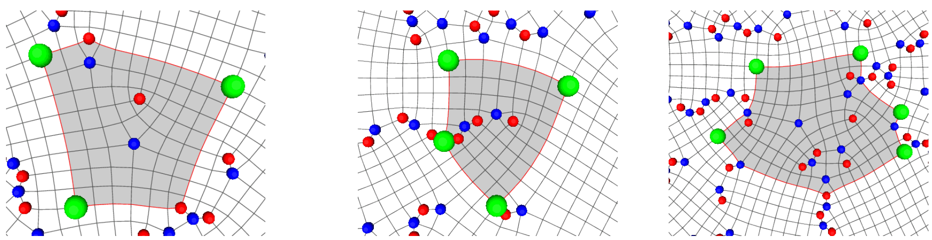


Figure 1: Shaded quadrilateral elements define a patch.

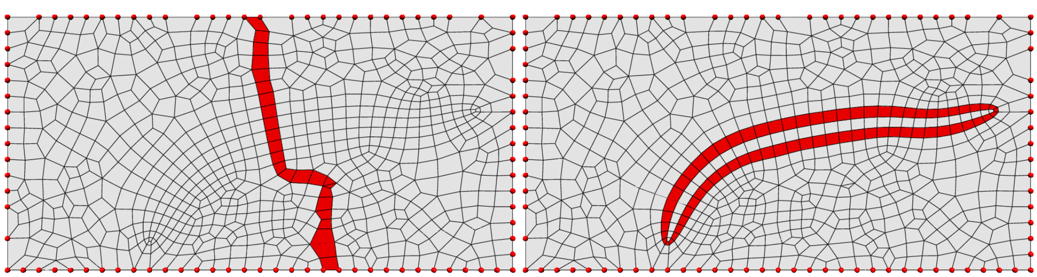
Definition 3. *A chord in a quadrilateral mesh is a set of quadrilateral elements formed by traversing opposite edges of a quadrilateral starting from any edge. There are two types of chords in a topologically valid quadrilateral mesh:*

1. **Boundary Chord:** *A chord which contains two boundary edges is called a boundary chord. In fact, in any topological valid quadrilateral mesh, any chord starting from a boundary edge must end at some other boundary edge (Figure 3a).*
2. **Cyclic Chord:** *If starting from an internal edge, traversal completes with the starting edge, then such a chord is called a cyclic chord (Figure 3b).*



(a) A 3-sided patch. (b) A 4-sided patch. (c) A 5-sided patch.

Figure 2: Examples of 3-5 and 5 sided patches.



(a) Boundary Chord. (b) Cyclic Chord.

Figure 3: Chords in a quadrilateral mesh.

Definition 4. *Let $\mathcal{S}$ be a smooth, orientable surface embedded in $\mathcal{R}^3$ and let $\mathcal{T}_p\mathcal{S}$ be the tangent plane at any point p on the surface. A cross-field f_p at the point p is defined as any cyclic set of four vectors $(u, u^{\perp}, -u, -u^{\perp})$ on $\mathcal{T}_p\mathcal{S}$ such that $|u| = 1$ (Figure 4)*

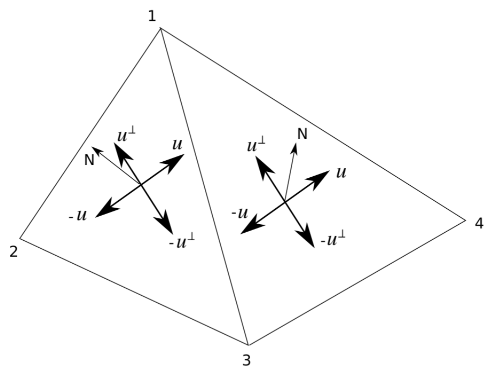


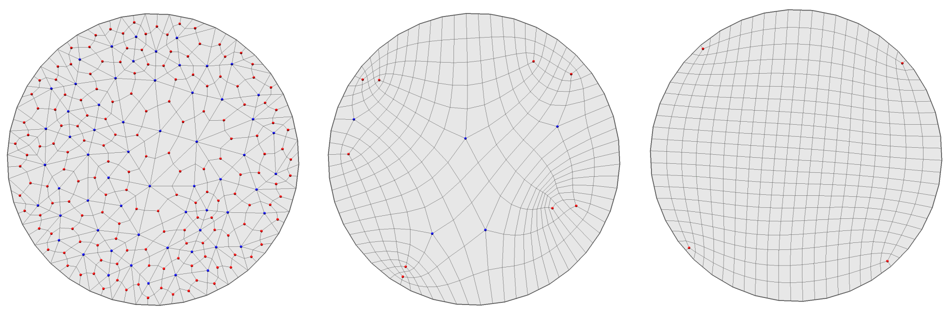
Figure 4: On a 3D surface each point is assigned four cyclic vectors.

3. Related Work

Mesh adaptation has been extensively studied since the beginning of mesh generation. Since this paper unifies all three adaptation techniques, i.e. improvement, refinement, and simplification, we cover all of them in this brief survey.

- **Mesh improvement:** A high-quality mesh is characterized by both geometric and topological qualities. Geometric qualities include element aspect ratio, area, min/max angle, etc. A complete list of various quality metrics is provided in the *Verdict* [27] software, and a thorough analysis of various metrics is presented by Shewchuck [32]. In geometric improvement, mesh nodes are repositioned to locations which optimize user-specified objective functions. Since the literature on mesh geometric optimization is vast, we refer the reader to *Mesquite* [6].
  For topological quality, we consider the degree of each node, and topological improvements involve modifying edges (through swapping, collapsing etc.) so that a mesh achieves a higher topological quality. In

[36], we proposed an algorithm based on *Minimum Singularity Template* (MST) to reduce the number of singularities in a quad mesh in localized regions, while maintaining geometric quality. Figure 5a illustrates an example of a quadrilateral mesh with a large number of singularities; after applying the standard MST algorithm, a mesh with significantly fewer singularities is obtained (Figure 5b). Although MST is effective in reducing singularities, both the number and placements of singularities may be sub-optimal. A significantly improved patch obtained with the method proposed in this paper is shown in Figure 5c.



(a) Input model. (b) MST result. (c) Ideal placement.

Figure 5: With the standard MST, quad mesh improvement is sub-optimal.

- **Mesh refinement:** Mesh refinement involves adding new elements in specified regions. As shown in Figure 6 a quadrilateral element can be refined into any number of smaller quadrilateral elements using recursion, but such a subdivision will always lead to additional singularities unless the boundary is also refined (the latter is not preferred in general). Schneider [31] proposed 2-refinement and 3-refinement templates as shown in Figure 7; an improved version was proposed by Garmella [15]. These templates are applied to the elements identified (grey elements in Figure 7) and tagged for refinement. In order to keep the mesh consistent, neighbouring elements must also be refined; and to avoid refining the entire mesh, singularities are inserted as illustrated. Although these templates lead to high geometric quality, they are hierarchical and produce many singularities in the region adjacent to the selected regions. In addition, application of some of these templates may create unstable refinement [31].

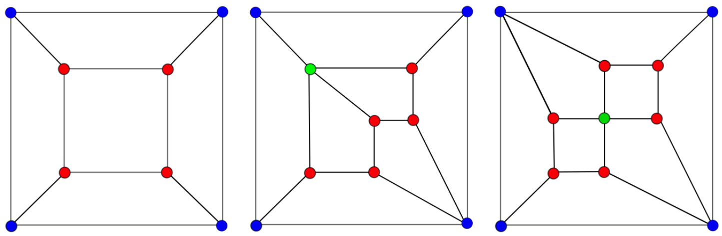


Figure 6: Local refinement of a quadrilateral element.

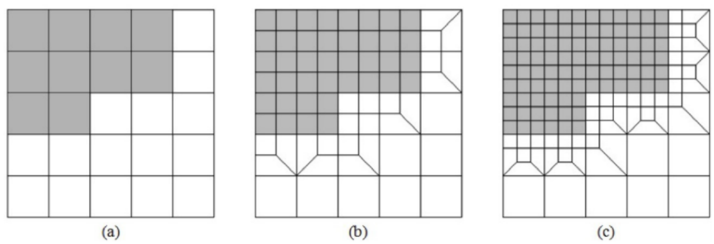


Figure 7: Schneider's templates can not refine arbitrarily without producing large number of singularities (Schneider [31]).

- **Mesh simplification:** Simplification involves deletion of elements until a prescribed threshold is achieved. For a quadrilateral mesh, simplification is far more challenging than improvement and refinement. For mesh simplification, many local operations such as quad-close [21], quad-collapse [9], edge split, vertex rotation [35], edge-flips, and quad-vertex merge [10] have been developed. Unfortunately, all of these operations increase singularities when applied to a patch containing a single singularity. To be effective, these operations must be applied to large regions [5], or a higher level structure within a quad mesh. *Poly-chord* is one of the structures, which has been exploited for quad simplification. In a poly-chord collapse, an entire line of quads within a chord is removed [4] (see Figure 8). Staten et.al. [11] showed that removal of cyclic chords produces localized coarsening. They also showed a way to create cyclic chords by stitching partial chords using local operation. Dewey et.al [11] later developed coarsening rings (within the coarsening region) and simplified the mesh by collapsing them. Although, it is simple to extract poly-chords passing through a region, applying them for the simplifications is usually non-trivial since: (1) a poly-chord encapsulates global structure and it can be arbitrarily complex; it can be self-touching, self-intersecting, and can span significant number of elements of the mesh, (2) if there are multitude of chords passing through a region, each one must be incrementally extracted and collapsed, (3) since a chord can extend beyond the localized region, it must be split into smaller independent parts, and (4) collapsing a chord may increase the degree of some nodes, therefore, local face-open operations must be applied after the chord operation.

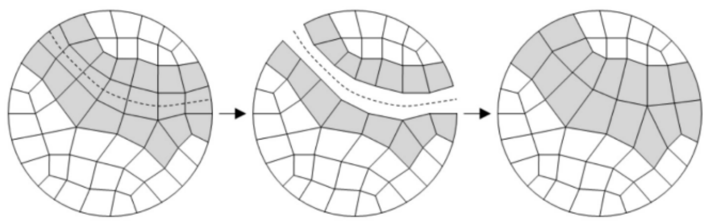


Figure 8: Quad simplification using chord removal (Anderson [1]).

To the best of our knowledge, only Tarini et.al [35] considered all three adaptation techniques in their work. They defined three kinds of local operations: *coarsening operations,* to simplify the mesh; *optimizing operations*, which change local connectivity without affecting the number of elements; and *cleaning operation*, which resolve invalid configuration. Similarly, Kinney [21] provided a large number of templates (more than 1000) for quadmesh clean-up.

3.1. Tools to adapt 3D quadrilateral meshes

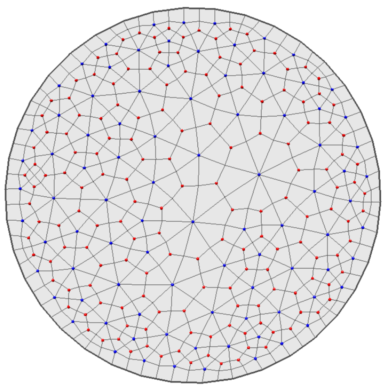
Using well-known tools, we can extend the method proposed here to adapt a quadrilateral mesh on 3D surfaces. For completeness, we briefly review these tools:

- **Surface Parameterization:** The parameterization of a surface involves bijective mapping between a 3D surface patch and a suitable 2D planar patch. Parameterization has been extensively studied in pure mathematics and computer graphics. We refer users to the comprehensive survey by Floater et al. [14]. In this paper, we have specifically used implementation of *Low Stretched Parameterization* from [37] and *Least Square Conformal Parametrization* from [22].

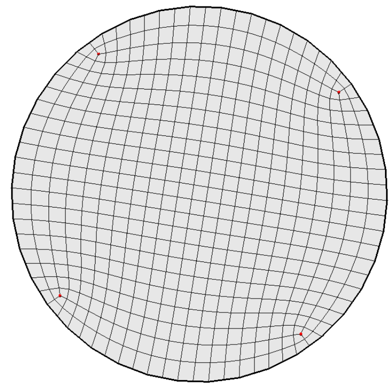
- **Mesh Clustering:** Mesh Clustering (or Mesh Partitioning) is an algorithm that partitions the faces of a mesh into non-overlapping and simply connected regions; see [13, 7]; many high-quality open-source libraries are available for mesh partitioning (examples: Metis[20], spectral clustering [33] etc.). The quality of the decomposition, however, is often application dependent, and these libraries allow users to define objective criteria such as roundness, planarity etc. In this paper, we use Metis software and iterative face clustering algorithm [19] to decompose a 3D surface mesh.
- **Cross Fields:** Many complex models can be effectively illustrated through non-photorealistic rendering method. For this reason, cross-fields were developed by Hertzmann and Zorin [17]. Symmetric vector fields on surfaces were developed by Palacious and Zhang [26] and they are known as N-RoSy fields. Crane etc [8] developed a method to evolve surfaces with directional control using a number of user defined singularities. Conjugate cross-fields [24] were developed to design planar quadrilateral meshes over architectural shapes. We shall rely on cross fields to orient the quad mesh on 3D surfaces.

4. Synopsis and Contributions

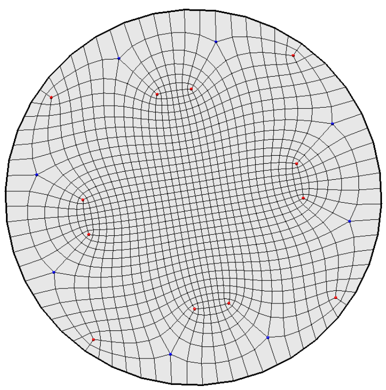
In this paper, we are proposing a method to adapt a quadrilateral mesh in specified regions. Typically, these regions are automatically identified and then tagged for refinement, simplification, or improvement. We also assume that these regions are disjoint, almost convex [23] and have disk topology. We refer to such a region as a *Patch*. Figure 9a shows one synthetic example of a patch. Observe that the input mesh has a large number of singularities. In addition, we also provide a single parameter α which controls the expected number of quad elements in the region. A value of $\alpha = 2.0$ indicates that the user expects double the number of elements in the patch after adaptation. Our proposed method adapts a patch with few singularities while maintaining high geometric quality. Figure 9 shows all three examples of mesh adaptation. While there are many approaches for each of these tasks, we will show that all these can be done very efficiently with one unified algorithm i.e. αMST.



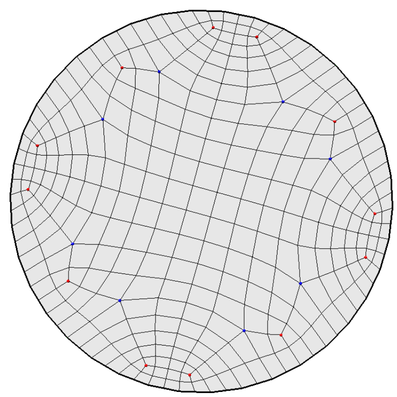
(a) Input mesh



(b) Improvement $\alpha = 1.0$



(c) Refinement $\alpha = 1.2$



(d) Simplification $\alpha = 0.80$

Figure 9: In α MST a single operation can perform improvement, refinement, and simplification.

5. Minimum Singularity Templates

In this section, we briefly describe our previous work [36] on the *Minimum Singularity Templates* (MST) and present the idea behind αMST templates.

The MSTs rely on the following four theoretical results from *Combinatorial Topology*:

(I) Every topological disk, with even number of boundary segments admits a quadrilateral mesh.

(II) Every polygon with k-sides has at-least $|k - 4|$ singularities [30].

(III) A single singularity cannot be moved, or removed from a quadrilateral mesh [28]

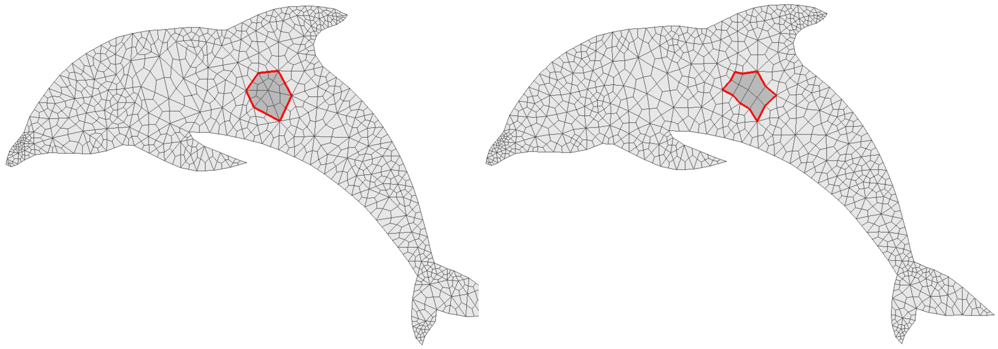
(IV) The minimum number of singularities in a domain is decided by its Euler characteristic, and it is invariant with respect to the geometric shape.

Statement II implies that 3-sided and 5-sided patches will always have at-least one interior singularity, and only a 4-sided patch can have zero interior singularity. Statement III implies that singularity modification requires at least two singularities. Statement IV places restriction on minimum number of singularities in a model. Based on these results, we presented a constructive algorithm [36] to generate low singularities templates for 3, 4, 5, and 6 sided polygons. These templates are called *Minimum Singularity Templates (MST)*. We refer readers to [36] for the complete analysis and procedure for applying these templates on any quadrilateral mesh.

5.1. Genesis of αMST

The standard MSTs are very effective in reducing singularities in a given patch (Figure 10). But they have limitations: for a given boundary segments on a patch, MST generates a new mesh with fewer number of singularities. However, these mesh templates do not have cyclic chords which we can exploit for mesh adaptation. Therefore, it is very challenging to adapt a MST patch without introducing additional singularities. These new singularities would allow adaptation and control of the maximum distortion of elements. Here we extend an N-sided MST by subdividing it into $N + 1$ sub-patches which will also introduce N singularities within the patch. Below we present the idea behind αMST using 4-sided patch, but this concept is applicable to other templates as well.

Figure 11a illustrates an abstract 4-sided αMST patch. Figure 11b shows a patch which we need to adapt. First, we extract its boundary as shown in Figure 11c and arrange the boundary edges in a counter-clockwise direction. Thereafter, we select four nodes from the boundary such that the opposite sides of a quadrilateral have same number of boundary edges (It is an essential condition for obtaining singularity-free quadrilateral elements in a patch.). These four nodes correspond to the four corners of the abstract patch. Now we move the four corners inside the domain and create one *core* and four 4-sided transitional patches surrounding the core. In addition, there are also four bridges which connect the core with the boundary of the patch.



(a) A patch containing singularities. (b) Patch after MST operation.

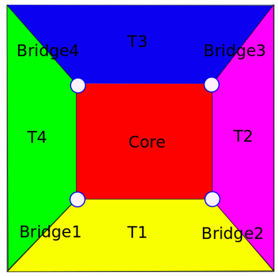
Figure 10: An example of standard MST operation applied to a patch.

In this manner, a patch is subdivided into five sub-patches. Improvement, refinement, and simplification are controlled by how the core is discretized. Each sub-patch is remeshed with the standard MST, therefore, we can provide a lower bound on the number of singularities in the patch. Figure 11d shows the results of mapping the template in the physical domain. To improve the quality of elements near the patch boundary, we apply *Mesquite* optimization and the result is shown in Figure 11e. Any modification in the core influences elements only in the core and the transitional patches; it has no effect outside the patch.

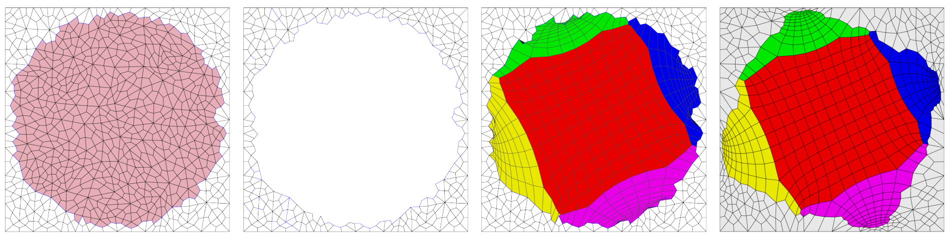
Figure 12 shows edge flows in both standard and αMST templates. An edge flow diagram indicates how the boundaries are split for generating quad mesh topology.

- **Improvement:** The boundary of the core is given the same number of nodes as the patch boundary (Figure 13a).
- **Refinement:** The boundary of the core is discretized with more nodes than the patch boundary (Figure 13b).
- **Simplification:** The boundary of the core is discretized with less nodes than the patch boundary (Figure 13c).

With this method, the core is discretized with desired number of quadrilateral elements and transitional sub-patches are discretized with a fewer number of singularities to accommodate all-quads elements. It should be noted that the number of singularities in each transitional patch is close to the theoret-



(a) An abstract patch.



(b) A patch in a quad mesh. (c) An empty patch. (d) Remesh the patch. (e) Optimize the patch.

Figure 11: The concept behind α MST.

ical least number, however the patch itself may have more singularities. In practice, after applying MST on each of the $N+1$ sub-patches, we also apply the standard MST over the entire patch and it usually eliminates additional singularities.

5.2. Generating cyclic chord

A cyclic chord provides many advantages in quadmesh adaptation [11]. When these chords are refined, they do not introduce new singularities and if they are simple, removing them is easy. Moreover, removal of a simple cyclic chord which is sandwiched between two simple cyclic chords also does not introduce new singularities. Unfortunately, such cyclic chords are rare in meshes generated with automatic quad mesh generators. With αMST we can generate them easily. Instead of refining the core of a patch, if we keep the number of nodes on the core boundary equal to the patch boundary (Figure 13d) and refine the bridges, then none of the sub-patches will have interior singularities. Such refinement of bridges will create cyclic chords in the transitional patches. Moreover, specifying the number of nodes on the

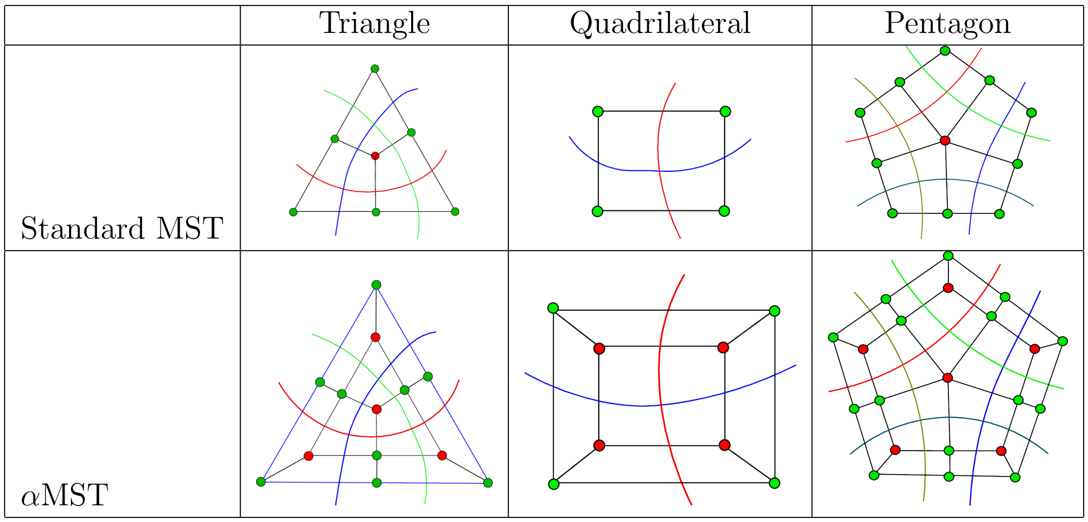
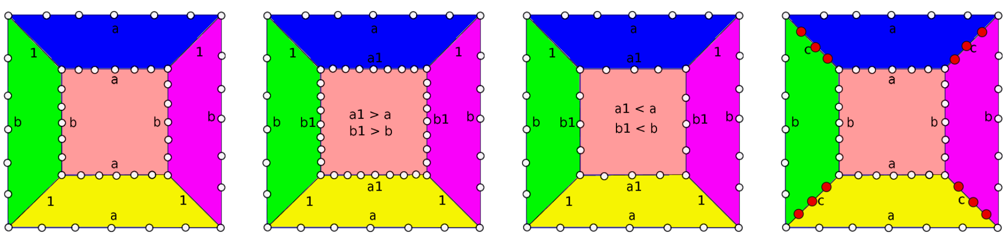


Figure 12: Edge flows in standard and αMST templates.



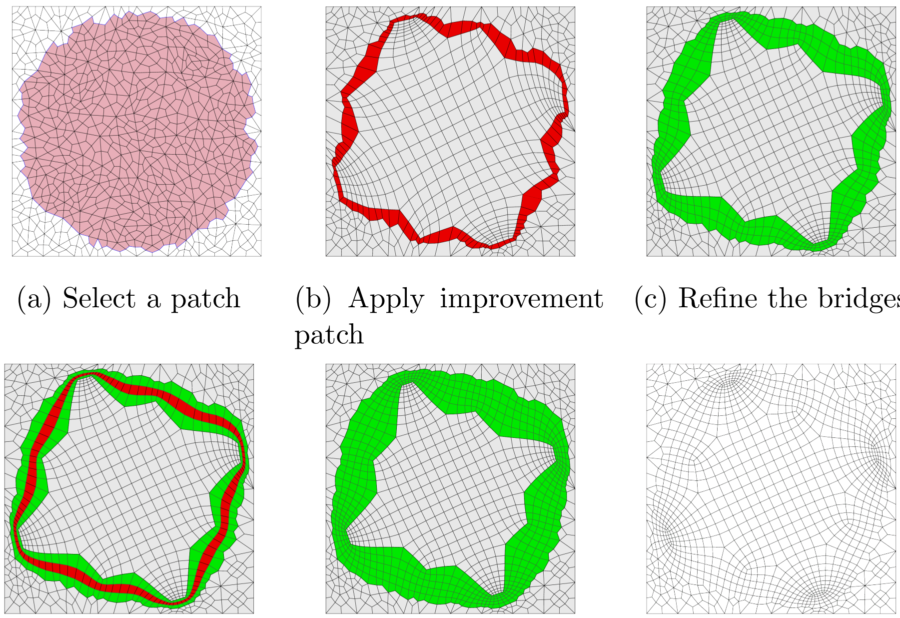
(a) Improvement. (b) Refinement. (c) Simplification. (d) Cyclic chords.

Figure 13: A single template for improving, refining, and simplifying a patch.

bridges is a free parameter, therefore, an unbounded number of cyclic chords can be generated with the αMST. Figure 14 shows the steps in generating cyclic chords. Figure 14f shows the mesh after Mesquite optimization. Such chords are the prime candidate for mesh refinement and simplification.

6. Applying templates

The αMST templates are flexible and can be applied automatically or interactively. During simulation, the user specifies regions where high density of elements are needed. Currently, our algorithm expects the region to be almost convex and have disk topology. The convexity allows the use of aggressive geometric optimization methods to improve the qualities of interior



(a) Select a patch (b) Apply improvement patch (c) Refine the bridges (d) Refining a cyclic chord. (e) Multiple cyclic chords. (f) Mesh after optimization.

Figure 14: Creating cyclic chords in a patch.

elements.

From a given patch our algorithm selects four corners on the boundary. As shown in Figure 15 these four corners determines the position of singularities in the patch. To reduce the total number of singularities from all the patches we should be able to merge singularities from the neighbouring patches without increasing them. Unfortunately, it is not easy to do unless every patch places these four nodes optimally. However, the optimal selection of these four corners is a complex problem and requires global optimization. For simplicity and without loss of generality, we pick the first node randomly and detremine remaining three nodes so that the patch produces no interior singularities and then apply αMST.

If the objective is to improve, refine, or simplify the entire mesh, then Figure 16b shows one way in which large circular patches are automatically selected using the medial axis of the domain. However, patches can also be created using Voronoi, or convex mesh decomposition algorithms.

After applying αMST to a patch, there may be many singularities on the periphery of the patch. Therefore, after adapting all the patches, we apply standard MST operations to the final mesh to reduce the singularities and then perform global optimization using Mesquite software.

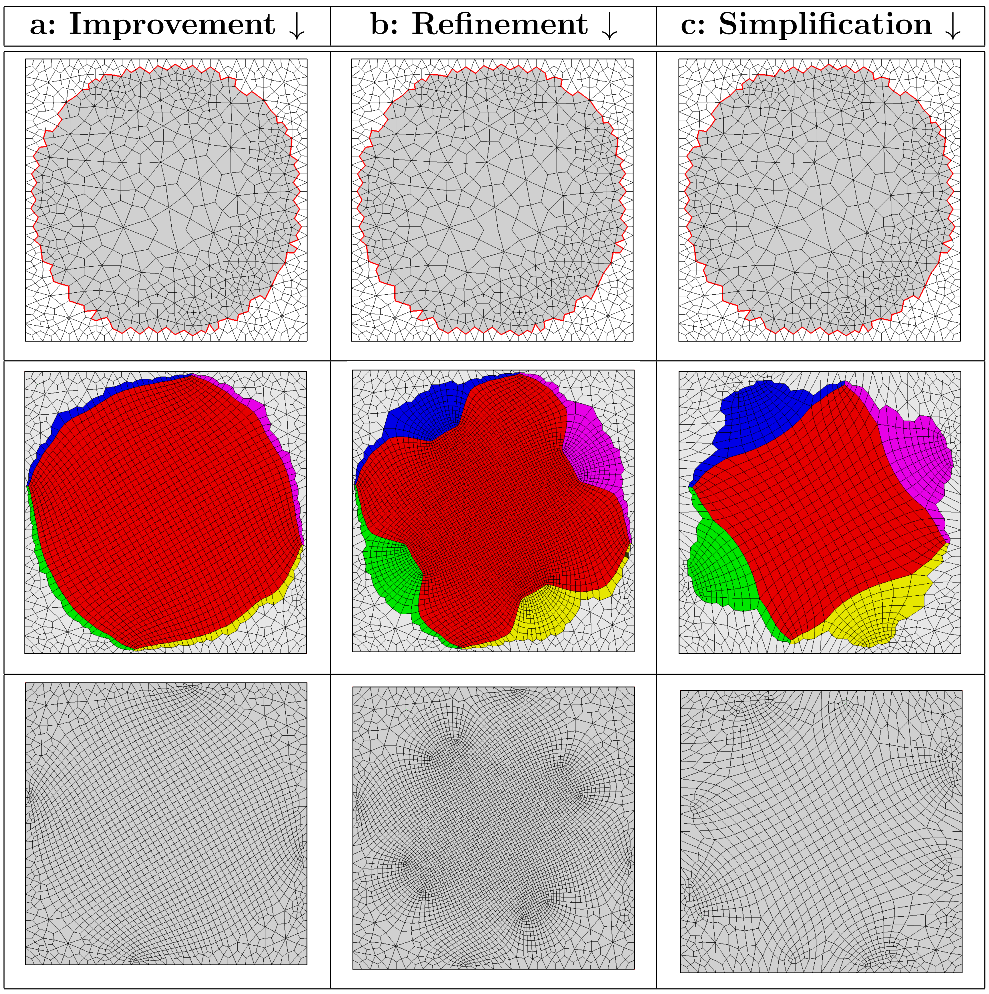
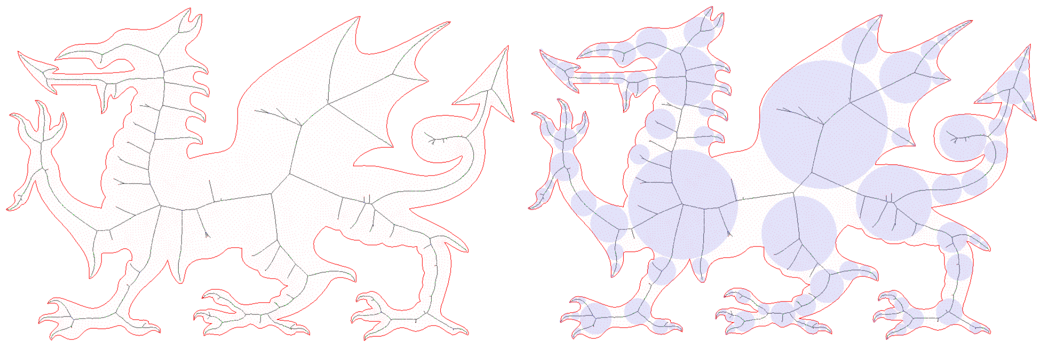


Figure 15: A simple example of αMST applied to a disk.

7. Surface Quadrangulation with aligned α-MST template core

The α-MST templates are combinatorial and do not depend on how patches are embedded in space. Therefore, they can be easily applied to 3D surface quad meshes. However, while applying these templates in 3D, two additional issues must be considered:



(a) The medial axis (b) Medial Circles

Figure 16: Medial circles can be used as patches.

1. The quadrilateral elements must not overlap (this could occur in regions of high curvature).
2. The mesh edges should preferably be aligned along the principal curvature directions.

The first issue can be resolved using classical surface parameterization techniques. With this method, we decompose the 3D geometric model into a small number of patches that can be mapped to a 2D disc. There is significant literature and many open-source software to decompose a model into compact topological disks (i.e. Euler characteristic = 1). In addition, these disks must allow bijective mapping between 3D and 2D space. Figure 17 shows one example in which a model is decomposed into such charts using Metis graph partitioning algorithm and then each patch is mapped to UV space. Such mappings are not unique and thefore different methods have distinct characteristics. In our experiments, we have used shape preserving mappings [37] and least square conformal mappings [22]. A open-source *libigl* [18] software provides implementations of these algorithms.

For the second issue, there are known methods to calculate the principal curvature directions over a triangle mesh [29]. However, we use an implementation of N-RoSy field algorithm which is available in *libigl* library [18]. An example of such a field is shown in Figure 18. In a well-structured quadrilateral mesh, alignment of mesh edges along the curvature lines is often preferred. Since very often not all quad edges can be perfectly aligned, this

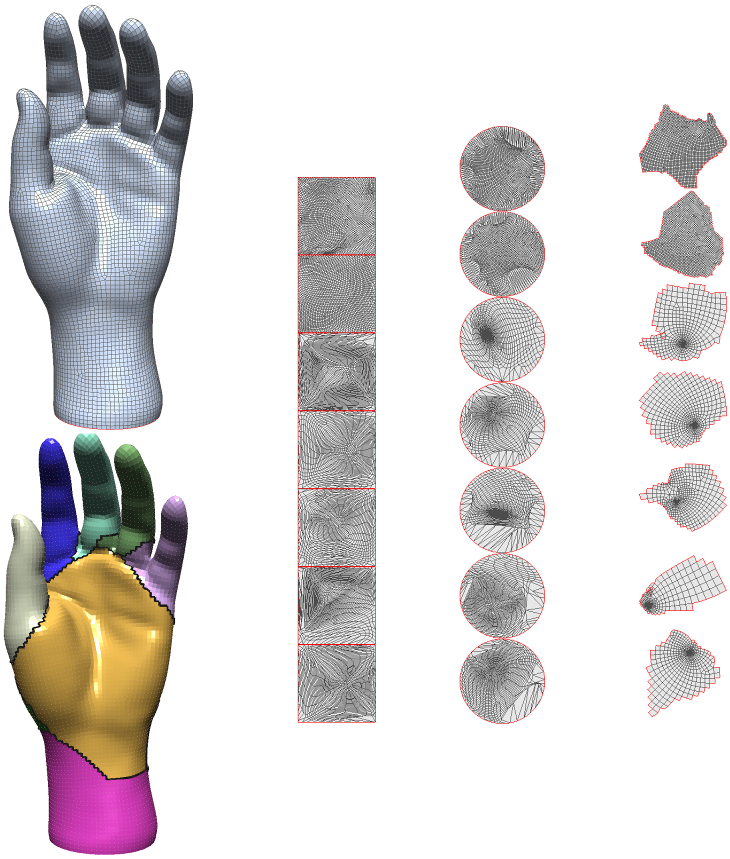


Figure 17: αMST templates can be applied over flattened patches to improve surface quad mesh.

problem is posed as an optimization problem. Edges with strict alignment requirements are grouped under *hard constraints* and edges in the less critical regions are grouped under *soft constraints*.

Coincidentally α-MSTs have one big advantage–their core is completely regular. This can be exploited to align all the elements in the core in the most preferred direction (instead of aligning each element individually). This necessitates choosing appropriate size of patches and selecting their corners. Even though many algorithms allow a surface to be decomposed into a small number of topological disks [12], alignment requirements often requires a large number of patches. Since the corners play the most critical role in maximum distortion of elements in the physical space, we use a simple and inexpensive procedure to identify them. First, based on the dominant direc-

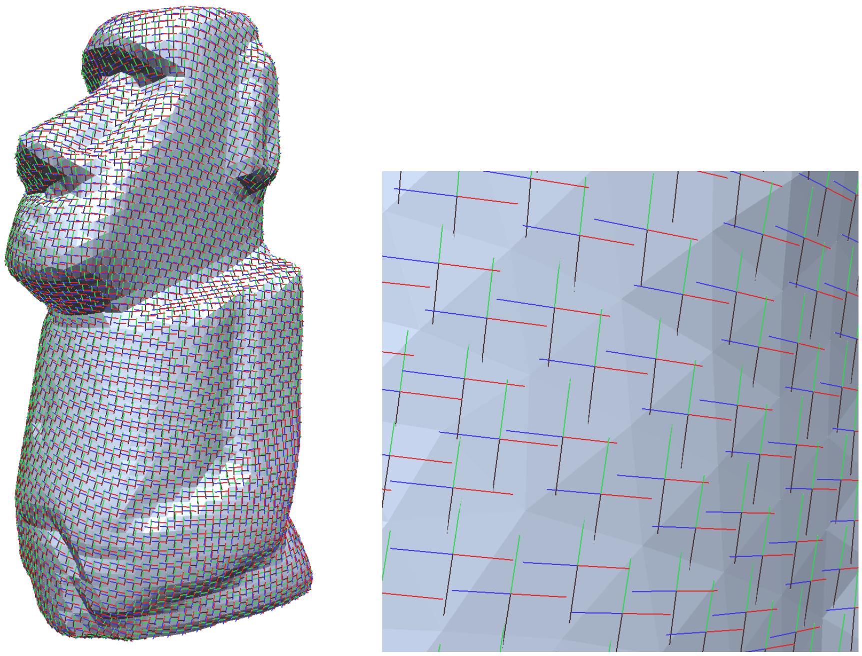


Figure 18: An example of 4-RoSy field over manifold

tion, we identify the lower left corner and incrementally move the corner in the clockwise direction if it improves the alignment quality. The remaining three corners are uniformly selected on the boundary of the patch. We iteratively select subgrids which have a large number of singularities or those patches in which singularities are clustered. Thereafter, we remesh these patches with αMST

It should be noted that the α value determines sampling of the surface ()e.g. $\alpha > 1$ would like the number of elements in a patch). However, it is not an exact multiplication factor. Pre-calculating the exact number of quad elements for a given α value could be useful which is a topic for our future work.

8. Results

1. **2D Results:** Figure 19a shows the example in which the input mesh has a large number of singularities. This input model was discretized with the Frontal algorithm of *Gmsh* software. As we can see in Figure 19b, mesh improvement significantly reduced the singularities. Table 1

shows the *Verdict* [27] mesh quality after each operation. Figure 19c shows the refinement with $\alpha = 1.5$. In this example, we refined only the large patches. Figure 19d shows the simplification with $\alpha = 0.80$. From these results, we observe that refinement and simplification do not increase the number of singularities and geometric qualities remain close to the input mesh. After simplification, there may be elements with high distortion. Such distortion can be minimized with better geometric optimization methods. In all these examples, we can notice that the mesh has a smooth transition from the adapted region to non-adapted region and singularities are well-spaced.

2. **3D Results:** We first compare our method with Tarini's datasets [35]. We evaluate the quality of simplification in terms of singularities and quality of alignment for the coarsest mesh. Tarini's quadrilateral mesh simplification results are shown in Figure 21. Our results are shown in Figure 22. We had to adjust the α value so that the number of nodes are comparable to Tarini's results. In all these experiments, our method produces significantly fewer singularities while maintaining the curvature flow lines. Tarini's results show that the meshes become unstructured when the simplification is performed with local operators. Since our method is non-local it is capable of capturing the topology structure from enlarged space and therefore it is capable of producing good quality quad mesh even at significantly decimated level.
   For the second example, Figure 23a illustrates a triangulated mesh containing 5M triangles, which we first quadrangulate and improve using MST (Figure b). Next, we carry out a series of mesh simplifications up 92% (Figures c-f). We observe that, the meshes are well aligned over a large scale. This substantiates our hypothesis that alignment of the core in the dominant direction is essential. It should also be noted that despite some distortion during 2D to 3D mapping,the absence of explicit boundaries in 3D permits aggressive global optimizations. The high quality 3D meshes obtained are a testimony of this claim.

9. Conclusion and future work

There are different methods available today for localized quadrilateral mesh adaptations, namely improvement, refinement, and simplifications. In this work, we have shown that all these can be carried out very efficiently

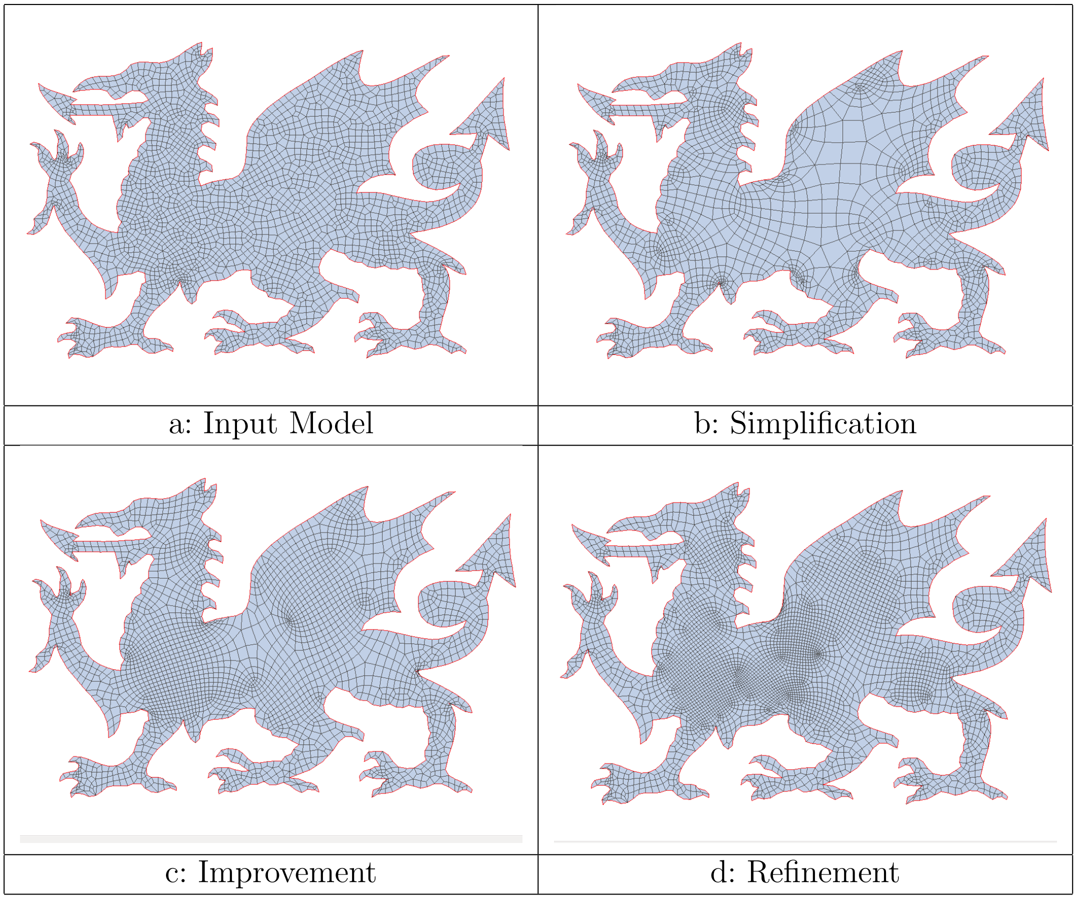


Figure 19: Mesh adaptation results using αMST method.

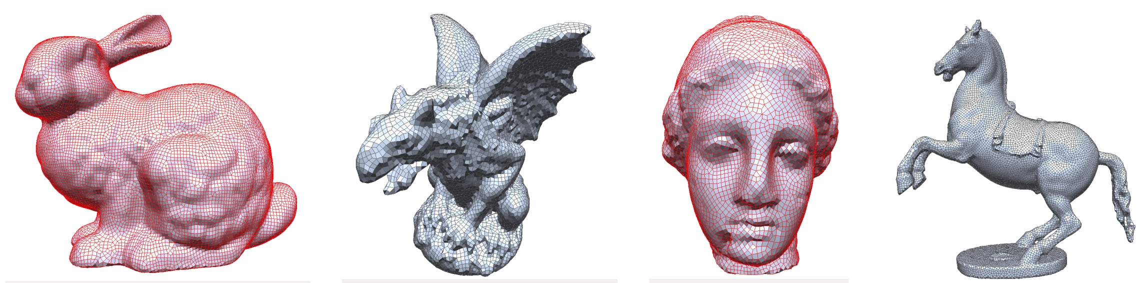


Figure 20: Input quadrilateral mesh for simplification methods comparisions.

using a single concept, namely αMST, where a single parameter α controls all three forms of adaptations. Unlike other methods, the proposed method

	Welsh			
	Input	**Improve**	**Refine**	**Simplify**
# Faces	7601	7789	12087	5612
# Singularities	572	278	602	400
Aspect Ratio	92.79%	96.20%	97.12%	83.48%
Condition Number	99.00%	99.75%	99.96%	99.56%
Distortion	98.10%	99.10%	99.19%	99.28%
MinAngle	98.80%	99.50%	99.82%	97.82%
MaxAngle	99.20%	99.80%	99.75%	96.00%
Jacobian	99.50%	99.92%	99.99%	99.91%
Scaled Jacobian	100.0%	100.0%	99.97%	99.26%
Shape	98.70%	99.10%	99.96%	99.64%
Shear	97.75%	99.25%	99.98%	99.57%
Skew	21.20%	23.78%	16.25%	41.66%
Taper	9.0%	10.0%	0.0%	0.0%
Warpage	99.90%	100.0%	99.99%	99.91%

Table 1: Mesh qualities reported by Verdict [27] software for the model in Figure 19

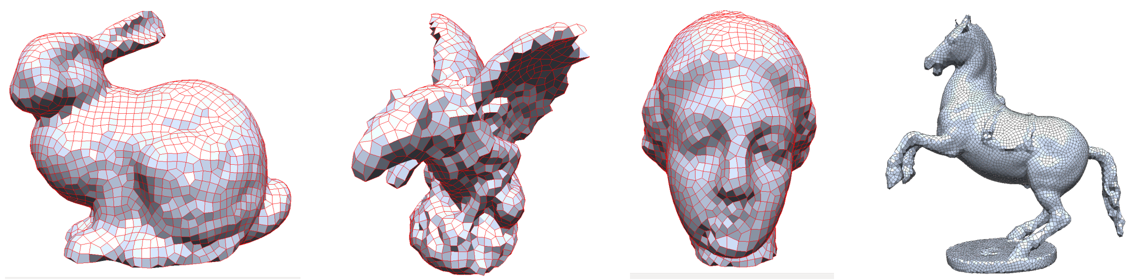


Figure 21: Quadrilateral meshes simplification using Tarini's method [35].

is non-hierarchical and stateless, yet it can arbitrarily modify any number of quadrilateral elements in a patch, while maintaining geometric quality. Moreover, our refinement and simplification processes are stable, i.e., applying these operations do not deteriorate the geometric quality of the elements. The simplicity of implementation and generality are additional advantages of our approach. Finally, all these operations are easy to unroll if the results do not match user's expectation. Simplification, in particular, is very attractive, as it does not use the dual-chords concept, which although simple, is not very intuitive. Our method also provides a deterministic control over the number

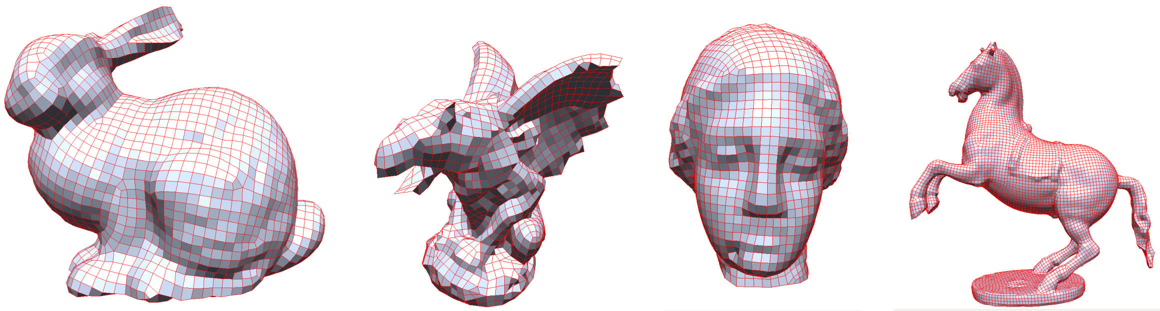


Figure 22: Quadrilateral meshes simplification using αMST.

Model	Input Quad Mesh	Tarini's Results	α-MST Results
Bunny	V11020/S4067	V3000/S956	V3008/S153
Gargoyle	V11104/S4283	V2060/S944	V2200/S242
Igea	V12038/S4205	V3044/S978	V2958/S82
Rampart	V20000/S6398	V10020/S3749	V10018/S452

Table 2: Simpification comparision results. Here $V\#$ is number of nodes and $S\#$ is number of singularities in the model.

of quadrilaterals and singularities.

The proposed templates are applicable to 3D surface quadrilateral meshes as well. Since alignment is an important criterion in 3D, we provide a simple method which aligns patches (and elements within) along dominant directions rather than aligning every element individually. In addition, since each patch is completely independent, the algorithm is linearly scalable.

Our method requires additional research in handling of narrow regions. Currently, in very narrow regions, singularities can cause high distortion. Anisotropic refinement and simplification are additional areas of research that we plan to explore. For 3D, chart size is ad-hoc and sometimes, users have to modify the patches interactively. For complex models, we need to explore better ways to identify charts which are computationally efficient and give guaranteed mesh quality.

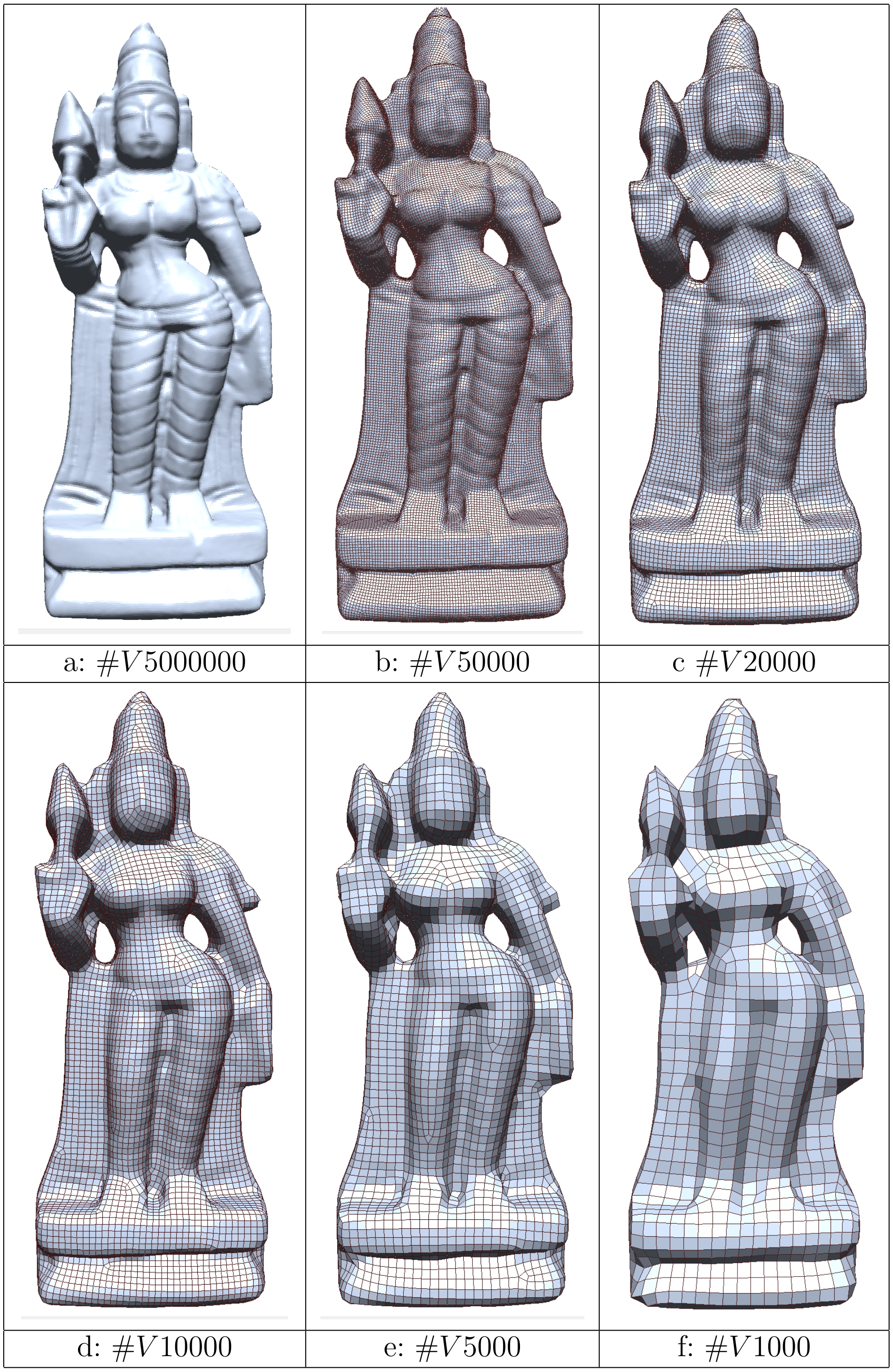


Figure 23: With α-MST isotropic elements with approximate alignment span over a large scale.

References

[1] Bret D. Anderson, StevenE. Benzley, and StevenJ. Owen. Automatic all quadrilateral mesh adaption through refinement and coarsening. In *Proceedings of the 18th International Meshing Roundtable*, pages 557–574. Springer Berlin Heidelberg, 2009.

[2] David Bommes, Timm Lempfer, and Leif Kobbelt. Global structure optimization of quadrilateral meshes. In *Computer Graphics Forum*, volume 30, pages 375–384. Wiley Online Library, 2011.

[3] David Bommes, Bruno Lvy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, and Denis Zorin. Quad-mesh generation and processing: A survey. *Computer Graphics Forum*, 32(6):51–76, 2013.

[4] Michael J. Borden, Steven E. Benzley, and Jason F. Shepherd. Hexahedral sheet extraction. In *Proceedings of the 11th International Meshing Roundtable, IMR 2002, Ithaca, New York, USA, September 15-18, 2002*, pages 147–152, 2002.

[5] Agostino Bozzo, Daniele Panozzo, Enrico Puppo, Nico Pietroni, and Luigi Rocca. Adaptive quad mesh simplification, 2010.

[6] Michael L. Brewer, Lori Freitag Diachin, Patrick M. Knupp, Thomas Leurent, and Darryl J. Melander. The mesquite mesh quality improvement toolkit. In *IMR*, 2003.

[7] Nathan A. Carr, Jared Hoberock, Keenan Crane, and John C. Hart. Rectangular multi-chart geometry images. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006.

[8] Keenan Crane, Mathieu Desbrun, and Peter Schrder. Trivial connections on discrete surfaces. *Computer Graphics Forum*, 29(5):1525–1533, 2010.

[9] Joel Daniels, Cláudio T. Silva, Jason Shepherd, and Elaine Cohen. Quadrilateral mesh simplification. *ACM Trans. Graph.*, 27(5):148:1–148:9, December 2008.

[10] Joel Daniels, II, Claudio T. Silva, and Elaine Cohen. Localized quadrilateral coarsening. In *Proceedings of the Symposium on Geometry Processing*, SGP '09, pages 1437–1444, Aire-la-Ville, Switzerland, Switzerland, 2009. Eurographics Association.

[11] Mark W. Dewey, Steven E. Benzley, Jason F. Shepherd, and Matthew L. Staten. *Proceedings of the 17th International Meshing Roundtable*, chapter Automated Quadrilateral Coarsening by Ring Collapse, pages 93–105. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[12] Jeff Erickson and Sariel Har-Peled. Optimally cutting a surface into a disk. *Discrete & Computational Geometry*, 31(1):37–59, 2004.

[13] Fengtao Fan, Fuhua (Frank) Cheng, Conglin Huang, Yong Li, Jianzhong Wang, and Shuhua Lai. Mesh clustering by approximating centroidal voronoi tessellation. In *2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling*, SPM '09, pages 301–306, New York, NY, USA, 2009. ACM.

[14] Michael S. Floater and Kai Hormann. *Surface Parameterization: a Tutorial and Survey*, pages 157–186. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[15] Rao Garimella. *Proceedings of the 18th International Meshing Roundtable*, chapter Conformal Refinement of Unstructured Quadrilateral Meshes, pages 31–44. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[16] Mark Halstead, Michael Kass, and Tony DeRose. Efficient, fair interpolation using catmull-clark surfaces. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, pages 35–44, New York, NY, USA, 1993. ACM.

[17] Aaron Hertzmann and Denis Zorin. Illustrating smooth surfaces. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 517–526, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[18] Alec Jacobson, Daniele Panozzo, et al. libigl: A simple C++ geometry processing library, 2016. http://libigl.github.io/libigl/.

[19] A. D. Kalvin and R. H. Taylor. Superfaces: polygonal mesh simplification with bounded error. *IEEE Computer Graphics and Applications*, 16(3):64–77, May 1996.

[20] George Karypis and Vipin Kumar. Metis – unstructured graph partitioning and sparse matrix ordering system, version 2.0. Technical report, 1995.

[21] Paul Kinney. Cleanup: Improving quadrilateral finite element meshes. In *6th International Meshing Roundtable*, pages 449–461, 1997.

[22] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.*, 21(3):362–371, July 2002.

[23] Jyh-Ming Lien and Nancy M. Amato. Approximate convex decomposition of polygons. *Computational Geometry*, 35(12):100 – 123, 2006.

[24] Yang Liu, Weiwei Xu, Jun Wang, Lifeng Zhu, Baining Guo, Falai Chen, and Guoping Wang. General planar quadrilateral mesh design using conjugate direction field. *ACM Trans. Graph.*, 30(6):140:1–140:10, December 2011.

[25] S. J. Owen, M. L. Staten, S. A. Canann, and S. Saigal. Q-morph: an indirect approach to advancing front quad meshing. *International Journal for Numerical Methods in Engineering*, 44(9):1317–1340, 1999.

[26] Jonathan Palacios and Eugene Zhang. Rotational symmetry field design on surfaces. *ACM Trans. Graph.*, 26(3), July 2007.

[27] Philippe P. Pebay, David Thompson, Jason Shepherd, Patrick Knupp, Curtis Lisle, Vincent A. Magnotta, and NicoleM. Grosland. New applications of the verdict library for standardized mesh verification pre, post, and end-to-end processing. In MichaelL. Brewer and David Marcum, editors, *Proceedings of the 16th International Meshing Roundtable*, pages 535–552. Springer Berlin Heidelberg, 2008.

[28] Chi-Han Peng, Eugene Zhang, Yoshihiro Kobayashi, and Peter Wonka. Connectivity editing for quadrilateral meshes. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, SA '11, pages 141:1–141:12, New York, NY, USA, 2011. ACM.

[29] Szymon Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In *Symposium on 3D Data Processing, Visualization, and Transmission*, September 2004.

[30] S. Schaefer, J. Warren, and D. Zorin. Lofting curve networks using subdivision surfaces. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '04, pages 103–114, New York, NY, USA, 2004. ACM.

[31] Robert Schneiders. Algorithms for quadrilateral and hexahedral mesh generation. *Proceedings of the VKI Lecture Series on Computational Fluid Dynamic, VKI-LS*, 4, 2000.

[32] Jonathan Richard Shewchuk. What is a good linear element? - interpolation, conditioning, and quality measures. In *In 11th International Meshing Roundtable*, pages 115–126, 2002.

[33] Horst D. Simon, Andrew Sohn, and Rupak Biswas. Harp: A dynamic spectral partitioner. *Journal of Parallel and Distributed Computing*, 50(12):83 – 103, 1998.

[34] Jos Stam. Flows on surfaces of arbitrary topology. *ACM Trans. Graph.*, 22(3):724–731, July 2003.

[35] Marco Tarini, Nico Pietroni, Paolo Cignoni, and Daniele Panozzo. Practical quad mesh simplification. *CG Forum (Eurographics*, pages 407–418, 2010.

[36] Chaman Singh Verma and Krishnan Suresh. 24th international meshing roundtable: A robust combinatorial approach to reduce singularities in quadrilateral meshes. *Procedia Engineering*, 124:252 – 264, 2015.

[37] Shin Yoshizawa, Alexander Belyaev, and Hans-Peter Seidel. A fast and simple stretch-minimizing mesh parameterization. In *Proceedings of the Shape Modeling International 2004*, SMI '04, pages 200–208, Washington, DC, USA, 2004. IEEE Computer Society.