5-6-7 Meshes

Nima Aghdaii* Hamid Younesy† Hao Zhang‡

Graphics, Usability and Visualization (GrUVi) Lab
School of Computing Science
Simon Fraser University

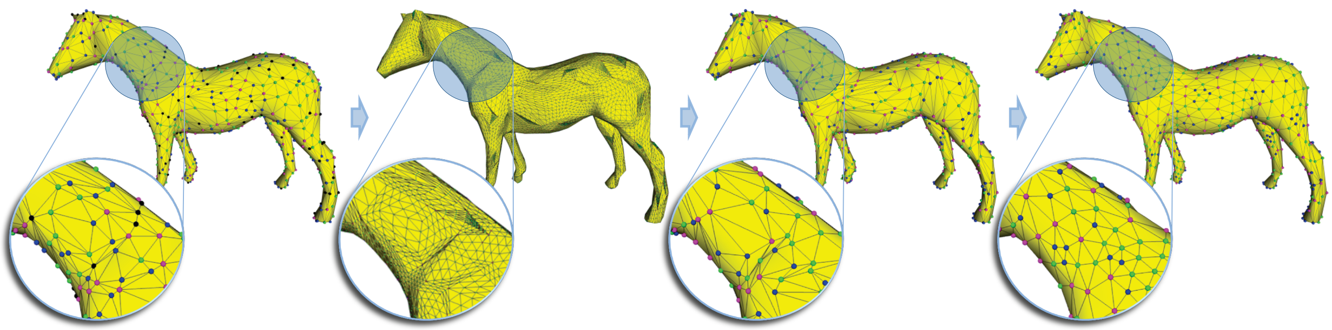

Figure 1: A triangle mesh (left) with low- and high-valence vertices (marked by black dots) remeshed into a 5-6-7 mesh with the same vertex count (right). Intermediate steps produce an initial 5-6-7 mesh (middle left) and a decimated version (middle right) which preserves the 5-6-7 property. The final mesh is produced with redistribution of vertices to improve sampling regularity, while respecting the features. Vertices of degrees 5, 6, and 7 are coloured by blue, green, and red, respectively.

Abstract

We introduce a new type of meshes called 5-6-7 meshes. For many mesh processing tasks, low- or high-valence vertices are undesirable. At the same time, it is not always possible to achieve complete vertex valence regularity, i.e., to only have valence-6 vertices. A 5-6-7 mesh is a closed triangle mesh where each vertex has valence 5, 6, or 7. An intriguing question is whether it is always possible to convert an arbitrary mesh into a 5-6-7 mesh. In this paper, we answer the question in the positive. We present a 5-6-7 remeshing algorithm which converts a closed triangle mesh with arbitrary genus into a 5-6-7 mesh which a) closely approximates the original mesh geometrically, e.g., in terms of feature preservation, and b) has a comparable vertex count as the original mesh. We demonstrate the results of our remeshing algorithm on meshes with sharp features and different topology and complexity.

Keywords: Geometry processing, Remeshing

1 Introduction

The valences of vertices in a triangle mesh often have an impact on how certain mesh processing algorithms perform. For example, valence-three vertices will cause an edge collapse operator to generate non-manifold vertices [8] and high-valence vertices can lead to visible artifacts in mesh subdivision [10]. When triangle quality is of concern, neither low- nor high-valence vertices are desirable since they often imply large or small face angles in a triangle mesh. It is commonly known that the regular vertex valence in a triangle mesh is 6 but complete regularity can be achieved only on a tessellation of genus-one surfaces. An intriguing question is whether it is always possible to completely eliminate low- and high-valence vertices, only keeping valences close to 6, e.g., 5, 6, and 7, for meshes tessellating surfaces of any arbitrary genus.

In this paper, we answer the above question in the positive. Specifically, we show that given an arbitrary closed triangle mesh with any genus, we can always remesh it to a *5-6-7 mesh*, i.e., a triangle mesh whose vertex valences only take on values 5, 6, or 7. We also show how to keep the face count comparable to the original mesh, while respecting features on the original mesh.

Our interest in the specific valences 5, 6, and 7 only is motivated by the Euler Characteristic formula, from which it can be shown that the average valence in a closed manifold triangle mesh is $6\,(1-\frac{(2-2g)}{n})$, where n is the number of vertices and g is the genus of the mesh. As such, by increasing the number of vertices, we will maintain an average valence of 6. However, since it is not always possible to have a mesh consisting of vertices of valence 6 only, vertices with valences higher than 6 and lower than 6 are generally inevitable. Thus the "next best scenario" in bounding the vertex valences away from the regular valence 6 would be to produce 5-6-7 meshes.

Our 5-6-7 remeshing algorithm works in two phases. First is the initial conversion which is guaranteed to convert an arbitrary closed mesh into a 5-6-7 mesh. This step keeps the changes to mesh geometry to minimal but will increase the face count and may produce uneven vertex distribution. In the second phase, the refinement phase, we perform mesh decimation and enhancement, while maintaining the 5-6-7 property.

- **5-6-7 remeshing:** We start by removing vertices with valence lower than 5, without introducing *geometric error*. By *geometric error*, we refer to the distance of the vertices to the original surface. After that we apply a planar subdivision scheme, which does not change the geometry either, to push high valence vertices away from each other and surround each high valence vertex with an unshared set of regular vertices.

*e-mail:naghdaii@sfu.ca
†e-mail:hyounesy@cs.sfu.ca
‡e-mail:haoz@cs.sfu.ca

Then we remove vertices with a valence greater than 7 using only *local* remeshing. This operation may introduce some geometric error.

- **5-6-7 mesh refinement:** We perform a 5-6-7 preserving simplification to turn the mesh towards having its original vertex count. Then a relaxation step is applied both to improve the triangle qualities and to reduce the geometric error produced by decimation.

The first phase of our method changes the geometry only slightly at high valence vertices. However, in order to reduce the size (face count) of the 5-6-7 mesh back to the size of the initial mesh, we apply decimation and geometric enhancement which may change the shape geometry to some extent. For this, user only needs to specify a feature preservation threshold and the number of iterations to relax. Our implementation provides an interactive tool to assist the user in choosing a reasonable value.

2 Related Works

The quality of a surface mesh is crucial for two primary purposes: 3D visualization and numerical simulation. Therefore, there has been an abundance of remeshing algorithms proposed in the past two decades to improve the quality of a given mesh [2]. Some remeshing algorithms are based on improving the geometry of the mesh by redistributing the points on the underlying surface, e.g. [11] among many others. Other works look more into the mesh connectivity, e.g. [3, 7] and strive to reduce the degree variance of the connectivity graph by removing irregularities, or at least moving them to more appropriate positions. Isenburg et al. [9] showed that there is an intrinsic connection between the geometry and connectivity and they are not totally independent. Therefore, a low quality connectivity usually imposes a deficient geometry as well. Our work falls into the latter category and it introduces a new type of meshes, 5-6-7 meshes, aimed at valence regularity.

In applications involving terrain representations, the so-called 4-8 subdivison surfaces [13] are widely used to achieve a semi-regular representation of a surface. However, subdivision-based schemes modify the mesh globally and do not provide an easy control over the connectivity of the final mesh.

There are also works that rely on mesh parameterization. Some of them divide the mesh into patches, and others perform a global parameterization of the whole mesh, e.g. [1, 6] and after a resampling in the parameter domain, the new triangulation is projected back to the 3D space.The main drawbacks of these methods are their sensitivity to the specific parameterization used, the cutting area used for models that are not isomorphic to a disk, the inevitable distortion, and finally, parameterization methods are usually slow and sometimes their inefficiency makes them impractical. We take a direct approach to 5-6-7 remeshing and work only locally on the original mesh.

Many remeshing schemes also apply local adaptations on the mesh, where a series of local modifications, such as vertex splits or collapses and edge flips, are performed on the mesh, e.g. [5, 11, 12] among others that can be found in the survey [2]. While all these existing methods are capable of producing meshes with nice vertex valence distribution, e.g., the distribution is sharply concentrated near the regular valence 6, we are not aware of any work that would generate meshes with a guaranteed bound on the valences. The best such bound would be [5, 7], which is the goal we set to achieve in this paper. The 5-6-7 meshes we produce can be useful for connectivity editing in triangle meshes [7], where their editing operations operate on a 5-6-7 mesh.

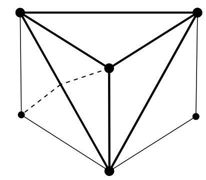

Figure 2: A simple scheme to remove a vertex of valence 3. A new vertex and two edges, dashed lines, are added to increase the degree of the v3 vertex by one.

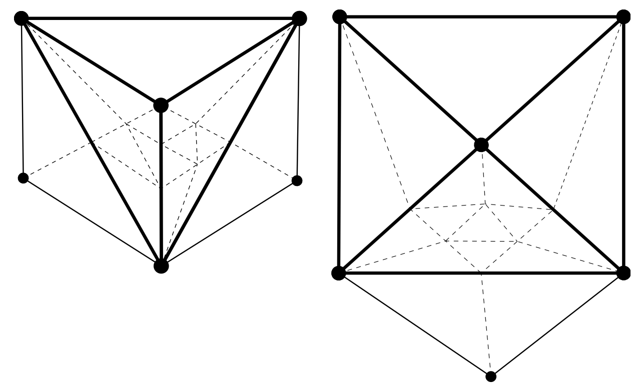

Figure 3: Elimination of a vertex with valence 3 (left) or 4 (right). Solid lines represent the edges in the original mesh and dashed lines are the new edges added by the remeshing process. Note that the original mesh edges are preserved and no geometry error is introduced.

3 5-6-7 Remeshing

In this section, we present a set of local remeshing schemes to convert a closed manifold mesh to a 5-6-7 Mesh. During this procedure, the size of the mesh in terms of face count is approximately multiplied by a factor of 10. Therefore, as described in the next section, a simplification process is applied to reduce the size of the final mesh, while preserving the 5-6-7 property.

We denote a vertex of valence d with vd and a vertex is called a $v567$ vertex if its valence is either 5, 6, or 7. We will refer to vertices with valence less than 5 as *low valence* and to vertices with valence greater than 7 as *high valence*.

3.1 Removing low-valence vertices

We start by removing all the low-valence vertices. During this step, all the v3 and v4 vertices are eliminated while the region outside of the 2-ring neighbourhood of each target vertex is kept intact.

A straightforward approach to removing a valence-3 vertex is to split an edge of one of its neighbouring faces and connect its midpoint to both this vertex and the vertex on the opposite side of the edge (Figure 2). This converts the v3 vertex into a v4 vertex and introduces a new v4 vertex, which has to be removed later. Instead, we replace any vertex of valence 3 or 4 by a 5-6-7 structure, as shown in Figure 3. These two structures can be adapted to any arbitrary geometry corresponding to a $v3$ or $v4$ vertex without introducing geometric error. This is beneficial, as in many cases, feature points of the mesh are vertices with a low valence and preserving the geometry around those is desirable. Also note that these structures only introduce new vertices that are $v567$ and may only

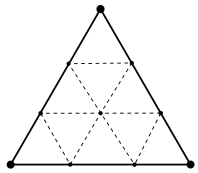

Figure 4: Topological subdivision to separate all the high-valence vertices sufficiently far apart from each other by at least two $v6$ vertices.

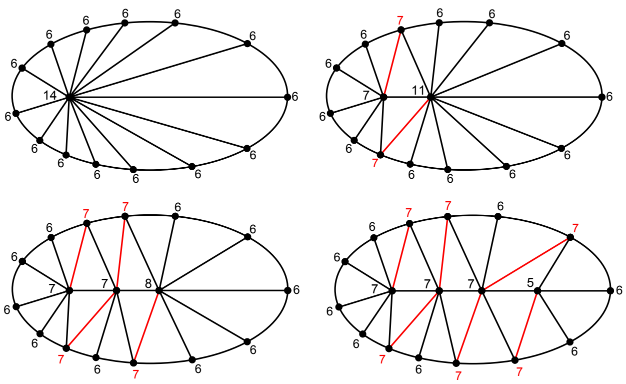

Figure 5: Elimination of a high-valence vertex ($v14$) by a series of vertex splits, where the ellipse denotes the one-ring neighbourhood of the vertex. This adds $\lfloor \frac{14-2}{3} \rfloor - 1 = 3$ of $v7$ vertices and one $v567$ vertex (in this example, the $v5$ vertex in the bottom-right image), while increasing the valence of the neighbour vertices by at most one (i.e. from $v6$ to $v7$).

increase the degree of some vertices in their two ring. But the valence of no vertex is decreased. As a result, no new $v3$ or $v4$ vertex is generated.

3.2 Subdivision

Before removing high-valence vertices, our method requires them to be far apart from each other in the connectivity graph. More specifically, every vertex with a valence higher than 7 should have a unique one-ring neighbourhood consisting of only regular vertices ($v6$).

This is often not the case. So, in order to guarantee this condition, we apply a planar subdivision rule to subdivide every face of the mesh into 9 faces, as shown in figure 4.

3.3 Removing high-valence vertices

In order to remove a high-valence vertex, we iteratively split it to v7 vertices until the remaining vertex has a degree less than or equal to 7. More specifically, we denote the vertex with degree > 7 as our *pivot*. Next, we replace the pivot with a vertex of degree 7 and another vertex of degree $deg(pivot) - 3$. The new vertex of degree $deg(pivot) - 3$ becomes our new pivot and we repeat the process until the degree of the pivot becomes less than or equal to 7. Figure 5 illustrates the iterations involved for removing a vertex of valence 14.

It can be shown that a vertex of valence h, where $h > 7$, can be replaced by $\lfloor \frac{h-2}{3} \rfloor - 1$ vertices with valence 7, and one v567 vertex, while increasing the valence of the vertices in the one-ring neighbourhood by at most one (refer to the appendix for the proofs).

Since the subdivision process has provided each high-valence vertex with a unique one-ring of regular vertices, the high-valence removal step replaces high-valence vertices with a 5-6-7 structure without introducing new high or low valence vertices.

In terms of positioning of the newly created vertices, although we can move all of them to the same location of the original high valence vertex and introduce no geometric error, the resulting mesh will have a degenerate geometry, which is not desirable. To solve this problem, we initially place all the newly created vertices at the position of the initial vertex, which is degenerate. Then we iteratively move each vertex towards the centroid of its adjacent vertices, while giving the vertex itself a higher weight to keep it close to its initial position. The new positions are iteratively calculated by the following equation:

$$u_{i_{new}} = \frac{\alpha u_i + \sum_{p \in N(u_i)} p}{\alpha + |N(u_i)|},$$

where u_i and $u_{i_{new}}$ are the positions of the i-th vertex before and after each iteration, $N(u_i)$ is the set of adjacent vertices for vertex u_i and α is a constant weight, which is chosen to be 50 in our implementation. Higher values of α keeps vertices u_i close to their original position and a value of zero moves them to the centroid of their neighbours. Since at each iteration two of the degenerate vertices are pulled away, for k newly added vertices, we require a minimum of $\lfloor \frac{k}{2} \rfloor$ iterations.

This positioning of the vertices might distort the mesh around u_i, which can be controlled by the value of α. However, the decimation and geometry enhancement process in the next step will move these vertices around in a geometry-aware manner. So, this equation is just used to create a non-degenerate mesh without fold-overs to start with.

It is worth noting that there are cases where it is inevitable to tolerate some error for the high valence removal step, unless if we move all the generated vertices to the location of the original vertex, resulting in a degenerate geometry that has no geometric error. For example, imagine a high valence vertex being in the same plane with its adjacent vertices. Then move every second adjacent vertex toward the direction of the face normal, to form a lemon reamer like shape. It can be seen that replacing the center vertex with a vertex of lower valence, will always introduce some geometry error.

4 Decimation and Enhancement

So far we have created a 5-6-7 mesh by slightly changing the geometry. However, this process has increased the face count of the mesh approximately by a factor of 10. In order to obtain the same face count back, we use a *5-6-7 preserving* simplification method to reduce the size of the mesh to as close as possible to its original size.

4.1 5-6-7 Preserving Mesh Decimation

We simplify the mesh using the edge collapse simplification, and we only allow those edge collapses, that still preserve the 567 property of the mesh. As shown in Figure 6(left), an edge collapse between vertices v_1 and v_2, with valences d_1 and d_2, creates a merged vertex of valence $d_1 + d_2 - 4$ (Lemma 1) and reduces the valence of vertices u_1 and u_2 by 1. Therefore, if either u_1 or u_2 has a valence of 5, we cannot collapse the edge. Moreover, we should maintain the condition $5 \le d_1 + d_2 - 4 \le 7$ or in other words, v_1 and v_2 can be either $v5$-$v5$ or $v5$-$v6$. We apply edge collapse mesh decimation governed by Quadric error measurement, while considering the 5-6-7 preserving constraints.

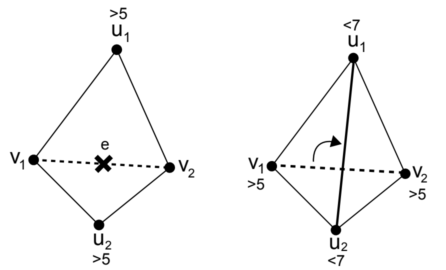

Figure 6: Edge collapse (left) and edge flip (right) that preserve the 5-6-7 property.

The simplification might stop at some point without any more possible edges to collapse. At this point, we iterate over all the edges of the mesh and mark those that create more collapsible edges, as shown in Figure 6 (right) and then we flip all of them. However, allowing an arbitrary edge to be collapsed may be dangerous and can result in a substantial error. Therefore we only allow an edge to be flipped if the dot product of the normal of its incident faces is beyond some threshold. We also perform the same check for the adjacent faces of the flipped edge. In our implementation a threshold of 0.9 is used. Note that increasing this threshold allows less edges to be flipped for the sake of a lower geometric error. One observation here is that we usually have many flips without any geometric error because the subdivision step generated many adjacent coplanar faces, for which flipping an edge between them will not introduce error.

The decimation process will alternatively decimate the mesh and then flip edges, until either the target face count is achieved or no more edges can be flipped.

4.2 Geometry enhancement

Although the decimated mesh maintains the 5-6-7 connectivity, the quality of the resulting mesh is not always desirable. Due to the edge flips and also because of the constraints on the quadric simplification, the decimation process is not able to collapse some of the low error edges and instead it has to consider the next candidate edges. Besides the geometric error, we observed that the decimation process decreased the quality of the triangles as well, by creating very small or long triangles. In order to address these issues, we apply an enhancement heuristic to relax the points on the surface of the original mesh, while respecting the geometry features.

More specifically, we apply a Laplacian smoothing followed by a back projection of the points to the surface of the original mesh. In order to project the points back to the original surface, we first find a mapping from the points of the decimated 5-6-7 mesh to the points of the original mesh, by considering the closest point from the original mesh to the point on the decimated mesh. Then, every point is projected back to the closest point on the one-ring neighbourhood of its corresponding vertex on the original mesh, which is the closest point to one of the triangles that is incident to its corresponding vertex.

After projecting the points back to the original mesh, we may need to update the correspondence that we calculated earlier, since the vertices are moving around. Let $c[u]$ be the corresponding vertex for u in the original mesh, we update $c[u]$ to the closest point from $N(c[u]) \cup \{c[u]\}$ to u, where $N(c[u])$ is the set of points in the one-ring of $c[u]$,

A drawback of this heuristic is that it smoothes out feature points after each iteration. To work around this, we detect feature points and fix their positions. We first examine the normals of every vertex's incident faces, and mark that as a feature vertex if there are two faces f and f' with $dot(\vec{n_f}, \vec{n_{f'}}) < \sigma_{threshold}$. The value of $\sigma_{threshold}$ is determined by the user. For instance, in Figure 7, a small positive value will work to fix the vertices on the edge of the cylinder. We also provide an interface to visualize the marked feature vertices as the user changes the value. A small value for $\sigma_{threshold}$ lets some of the feature points disappear but allows a better distribution of the points on the surface of the mesh and a larger value will fix more points, making the mesh less flexible towards changes.

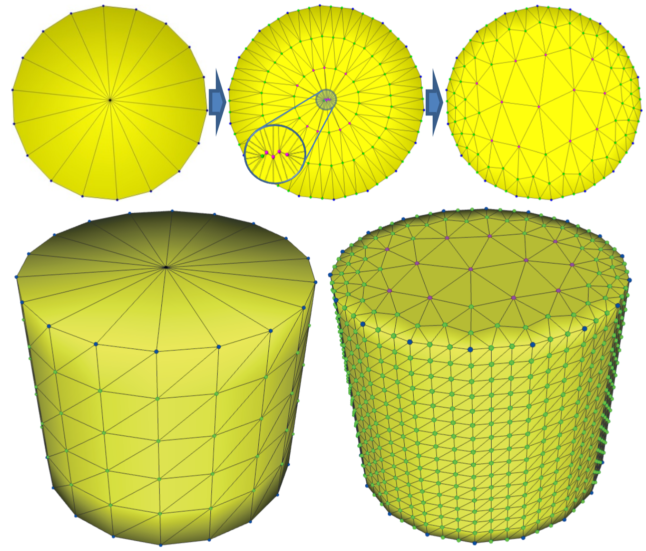

Figure 7: Comparison of the original cylinder mesh(left) with the 5-6-7 enhanced mesh(right). The top row from left to right: the original, 5-6-7, and 5-6-7 enhanced mesh (without decimation).

In cases where the increased size of the mesh is not an issue, for example when the initial mesh has a low face count, we can bypass the decimation step and run the geometry enhancement right after we create a 5-6-7 mesh. In such a case, the size of the mesh is increased but the process will become faster as the decimation step is the slowest step of our algorithm, and we also will not have the extra error added by the decimation step. Figure 7 illustrates a cylinder before and after converting it to a 5-6-7 mesh (without decimating) followed by the geometric enhancement step. So, the final mesh has significantly more faces than the original one but the high valence vertex is replaced by a set of 5-6-7 vertices, which are relaxed on the surface after the geometry enhancement.

5 Results

Our implementation of 5-6-7 remeshing takes a closed triangle mesh and turns it into a 5-6-7 mesh, which is simplified afterwards and then geometrically enhanced. Although the 5-6-7 remeshing has a linear time complexity, the decimation process can be fairly slow and take minutes to run on a mesh with 25K vertices.

We tested our algorithm on meshes with different topologies and different types of features. Figure 8 demonstrates the result of applying our method on a genus two manifold and Figure 9 exhibits our feature preservation mechanism. For the geometry enhancement, the user needs to set the rigidity factor ($\sigma_{threshold}$) and the number of iterations. Since picking the rigidity factor is not always intuitive, an interactive tool assists the user to pick a reasonable value.

We also incorporate edge lengths in quadric errors such that, shorter edges become better candidates to be collapsed. This attempts to eliminate small dense groups of vertices and tends to equalize the edge lengths of the mesh, which gives a more uniformly distributed vertex set.

In order to quantify the quality of the resulting mesh, we compute an approximation error measured by the well-known Metro tool [4], which calculates the Hausdorff distance between the original mesh and the final mesh after geometry enhancement. The error is mostly a result of the simplification process, and 5-6-7 remeshing itself does not add significant error to the mesh. The error as well as various other statistics related to our remeshing algorithm, such as the execution time, are shown in table 1.

6 Limitations and Future Works

The algorithm presented is an initial attempt at 5-6-7 remeshing and it still leaves room for improvement. One limitation is the increase of vertex count after the initial conversion. An immediate consequence of this is inefficiency when the input mesh has a large number of vertices. Most of these additional vertices come from the subdivision step, since we need to sufficiently separate the high-valence vertices from each other, to guarantee the 5-6-7 remeshing using only local rules. Perhaps more global remeshing operators that operate on larger regions of the mesh can alleviate the problem. In most of our experiments we observed that our decimation algorithm is able to decimate the mesh to its original size. However, that is not a guarantee. One counter example can be the tetrahedron, where there is no way to decimate a 5-6-7 version of a tetrahedron down to 4 vertices and keep the 5-6-7 property. Another interesting challenge is creating a 5-6-7 mesh with minimum number of v5/v7 vertices.

In the geometry enhancement process, since we find the initial correspondence by relying on the closest vertex, in cases where surfaces become to close geometrically, we may pick a wrong vertex on the opposite surface and project it incorrectly. Also, choosing the right parameter for fixing feature points affects the quality of the results. Using a small threshold may fix some vertices, allowing others around it to move and introduce a fold over. To work around this, we check for fold overs before moving a vertex. After all, the geometry enhancement step is a heuristic, prone to have some pathological cases such as cases with very close surfaces, or with geometric noise, which may be detected as features.

Geometry enhancement can also be implemented using an octree to locate the closest point on the other mesh efficiently, and projecting the point back to the one-ring neighbourhood of its corresponding vertex gives a fairly good approximate of the back projection to the mesh.

Currently, our remeshing algorithm is designed to work with closed manifold meshes only and the support for meshes with boundaries is left as a future work. Also, we have not investigated the applications of 5-6-7 meshes in this paper. We would like to investigate the utility of 5-6-7 meshes for mesh compression, subdivision, connectivity editing, and proving bounds on the angles of the triangles in the mesh. Another interesting question is the placement of valence 5 and 7 vertices. It may be desirable to place v5 vertices near convex regions and v7 vertices near concave regions, or considering the density of the points in order to correspond to the frequency of the mesh at that location. Finally, we recall many previous works which regularize mesh connectivity by producing meshes that are dominated by valence-6 vertices, e.g., [11]. It would be interesting to truly compare such meshes with 5-6-7 meshes in various relevant applications.

7 Conclusion

In this paper, we show that a closed triangle mesh with any genus can always be converted into a 5-6-7 mesh, a mesh with only valence-5, 6, and 7 vertices. The initial conversion scheme removes low- and high-valence vertices one by one and during the process, it creates a fairly large number of new vertices. We address this with a mesh decimation and geometry enhancement, while preserving the 5-6-7 property. In the end, we obtain a 5-6-7 mesh that closely approximates the original mesh, i.e. features are respected and has a comparable vertex count. However, It might not always be possible to decimate the mesh to its original face count, and, in some cases it is even theoretically impossible to decimate the mesh to its original face count (e.g. a tetrahedron or a great stellated dodecahedron).

A summary of our approach can be divided into the following four steps:

1. **Low Valence Removal:** v3 and v4 Vertices are removed efficiently without introducing error.

2. **Subdivision and High Valence Removal:** Vertices with valence greater than 7 are removed and an arbitrarily low error is introduced. An error of zero is achievable in a degenerate geometry. By the end of this step, we have a 5-6-7 mesh that has a low geometric error and the error only appears at high valence vertices. However, the size of the mesh is multiplied by approximately 10.

3. **Decimation:** In this step we decimate the mesh towards the initial face count as much as possible, while preserving the 5-6-7 property. This step is computationally expensive and might introduce a considerable error to the geometry.

4. **Geometry Enhancement:** We use a heuristic to improve the quality of the triangles, and projecting the mesh back to the surface of its original mesh. To preserve the features of the mesh, we detect them and fix their position. While this heuristic works in many cases, there are situations that it might generate a considerable error.

Acknowledgement

The authors would first like to thank the anonymous reviewers for their valuable feedback. Thanks should also go to the members of the Graphics Usability and Visualization (GrUVi) lab at Simon Fraser University (SFU) – specifically Ramsay Dyer, Steven Bergner, Oliver van Kaick and Ibraheem Alhashim for their fruitful discussions on the paper and the ideas therein. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), grant no. 611370 for Hao Zhang, and the PGS-D scholarship of the Natural Sciences and Engineering Research Council (NSERC).

References

[1] P. Alliez, M. Meyer, and M. Desbrun. Interactive geometry remeshing. *ACM Trans. Graph.*, pages 347–354, 2002.

[2] P. Alliez, G. Ucelli, C. Gotsman, and M. Attene. Recent advances in remeshing of surfaces. In *Shape Analysis and Structuring*, pages 53–82. 2008.

[3] Y. K. P. W. Chi-han Peng, Eugene Zhang. Connectivity editing for quadrilateral meshes. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)*, 2011.

[4] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, pages 167–174, 1998.

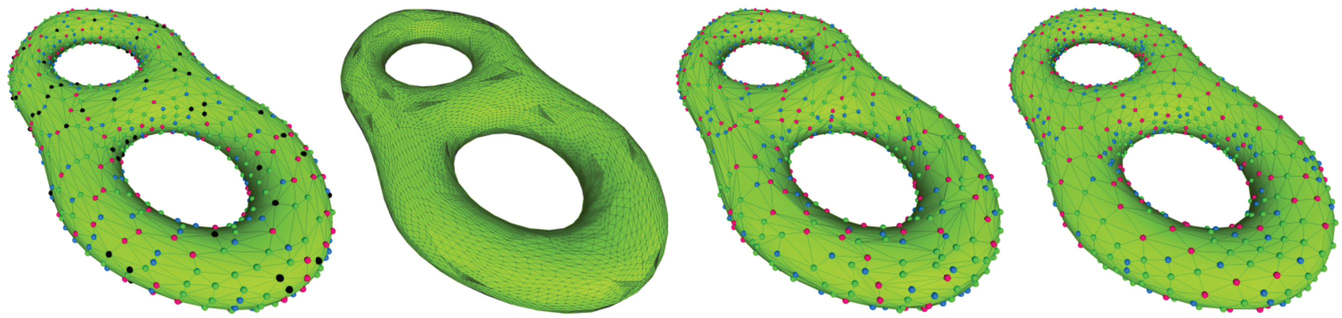

Figure 8: 5-6-7 remeshing on a genus 2 mesh. From left to right: the original mesh, the 5-6-7 mesh, the decimated 5-6-7 mesh and finally the enhanced 5-6-7 mesh.

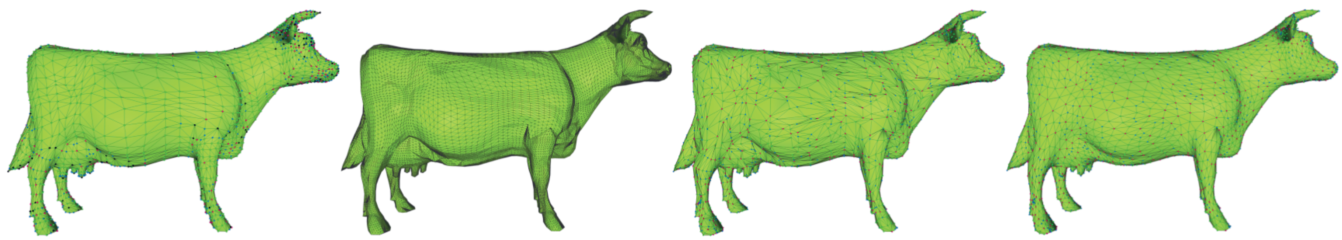

Figure 9: An example of respecting sharp features on a complex mesh. From left to right: the original mesh, the 5-6-7 mesh, the decimated 5-6-7 mesh and finally the enhanced 5-6-7 mesh.

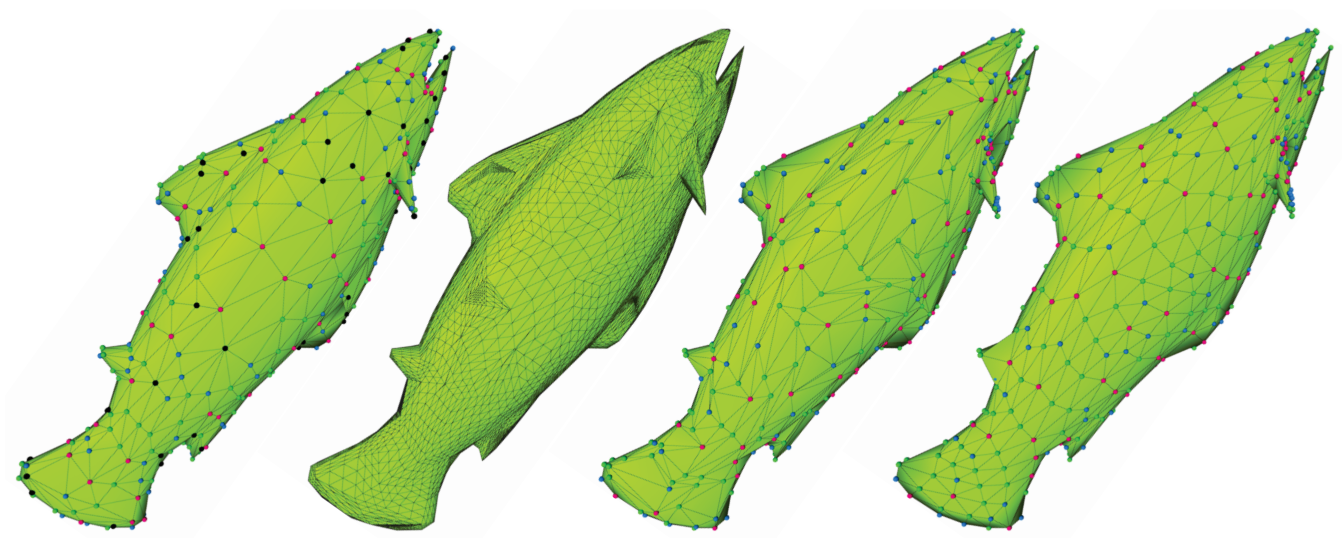

Figure 10: 5-6-7 remeshing of Fish mesh with fairly flat areas as well as some sharp feature regions. From left to right: the original mesh, the 5-6-7 mesh, the decimated 5-6-7 mesh and finally the enhanced 5-6-7 mesh.

Model	Metro error	#Faces	Valence					Time
			V_{low}	V_5	V_6	V_7	V_{high}	
fish	0.010	1K	8.9%	24.5%	36.1%	22.5%	8%	3.3s
horse	0.018	1.5K	7.7%	27.7%	33.6%	22.1%	8.9%	4.5s
venus	0.020	1.5K	9%	25.7%	35.6%	19%	10.7%	4.2s
eight	0.010	1.5K	5.1%	23.2%	42.6%	23.9%	5.2%	5.1s
cow	0.020	6K	3.2%	17.9%	61.7%	12.5%	4.7%	18.7s
armhand	0.007	25K	3.6%	26.2%	42.7%	22.2%	5.3%	13m

Table 1: Various statistics related to 5-6-7 remeshing on several meshes. Metro error is computed between the original and the final mesh. Time represents the total running time of the 5-6-7 remeshing algorithm and decimation. The number of vertices (in percentages) of different valences before remeshing is also included.

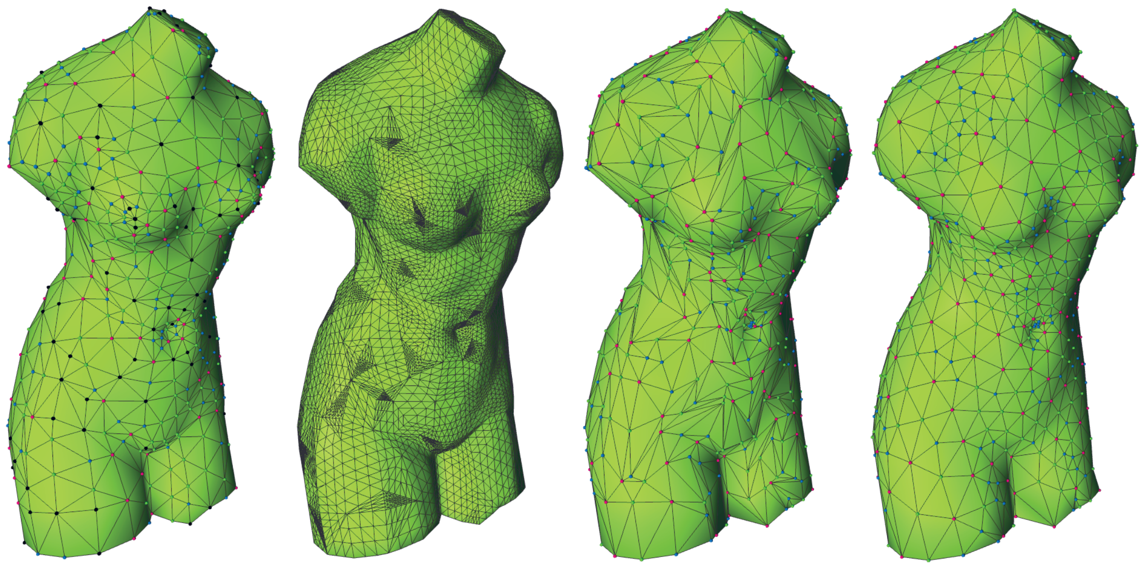

Figure 11: 5-6-7 remeshing of Venus mesh with subtle feature areas. From left to right: the original mesh, the 5-6-7 mesh, the decimated 5-6-7 mesh and finally the enhanced 5-6-7 mesh.

[5] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, pages 19–26.

[6] K. Hormann, U. Labsik, and G. Greiner. Remeshing triangulated surfaces with optimal parameterizations. *Computer-Aided Design*, pages 779–788, 2001.

[7] Y. Li, E. Zhang, Y. Kobayashi, and P. Wonka. Editing operations for irregular vertices in triangle meshes. In *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)*, pages 153–165, 2010.

[8] D. Luebke, B. Watson, J. D. Cohen, M. Reddy, and A. Varshney. *Level of Detail for 3D Graphics*. 2002.

[9] C. G. M. Isenburg, S. Gumhold. Connectivity shapes. In *In IEEE Visualization 2001 Conference Proceedings (2001*, pages 135–142, 2001.

[10] J. Peters and X. Wu. The distance of a subdivision surface to its control polyhedron. *J. Approx. Theory*, pages 491–507, 2009.

[11] V. Surazhsky and C. Gotsman. Explicit surface remeshing. In *Proc. Symp. on Geom. Processing (SGP)*, pages 20–30, 2003.

[12] G. Turk. Re-tiling polygonal surfaces. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 55–64, 1992.

[13] L. Velho. Using semi-regular 4-8 meshes for subdivision surfaces. *J. Graph. Tools*, pages 35–47, 2000.

APPENDIX

Lemma 1. *The result of merging two vertices of degrees d_1 and d_2 is a vertex of degree $d_1 + d_2 - 4$.*

Lemma 2. *On a closed manifold M with the connectivity graph G_M, let $V(G_M)$ be the vertex set of G_M and let $V' \subseteq V(G_M)$. Also, let ω be the result of merging V' into one vertex. The degree of ω is given by the following formula, if the subgraph G' induced by V' is a tree.*

$$deg(\omega) = \sum_{v'_i \in V'} deg(v'_i) - 4(|V'| - 1) \tag{1}$$

Proof. The proof can be done by induction over the size of the tree and using lemma 1. □

Lemma 3. *The remaining vertex from the splitting algorithm is a 5-6-7 vertex.*

Proof. By contradiction, suppose that in the last step of the algorithm we have a vertex of degree $y \geq 8$ which is split into a vertex of degree 7 and a vertex of degree $x \leq 4$. So,

$$y - x \geq 4 \tag{2}$$

Using lemma 1, we have $y = x + 7 - 4$ So,

$$y - x = 3 \tag{3}$$

(2) and (3) implies contradiction. □

Theorem 1. *Every vertex of degree h, $h > 7$, will be replaced by $\lfloor \frac{h-2}{3} \rfloor - 1$ number of $v7$ vertices and one 5-6-7 vertex.*

Proof. Using lemma 3, we know that the remaining vertex will be a 5-6-7 vertex. Now suppose that after the split procedure we have $|V'|$ vertices consisting of $|V'| - 1$ number of $v7$ vertices and one 5-6-7 vertex. By merging these vertices together we will recover the original high valence vertex, which had a degree of h. Let $h = deg(\omega)$ and let υ_{567} be the remaining vertex from the splitting algorithm. Now, using lemma 2 we have,

$$\begin{aligned} h &= \sum_{v'_i \in V'} deg(v'_i) - 4(|V'| - 1) \\ &= (7(|V'| - 1) + deg(\upsilon_{567})) - 4(|V'| - 1) \\ &= 3(|V'| - 1) + deg(\upsilon_{567}) \end{aligned}$$

$$|V'| = \frac{h - deg(\upsilon_{567}) + 3}{3} \tag{4}$$

$$= \frac{(h-2) - t}{3} \tag{5}$$

Where $t \in \{0, 1, 2\}$ and it makes $h - 2$ divisible by 3. So,

$$|V'| = \left\lfloor \frac{h-2}{3} \right\rfloor \tag{6}$$

Which is $|V'| - 1$ vertices of degree 7 and one 5-6-7 vertex. □