5-6-7 Meshes: Remeshing and Analysis

Nima Aghdaii[a], Hamid Younesy[a], Hao Zhang[a]

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

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, Graph connectivity

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 [1] and high-valence vertices can lead to visible artifacts in mesh subdivision [2]. 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. 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

Email addresses: naghdaii@sfu.ca (Nima Aghdaii), hyounesy@sfu.ca (Hamid Younesy), hoaz@sfu.ca (Hao Zhang)

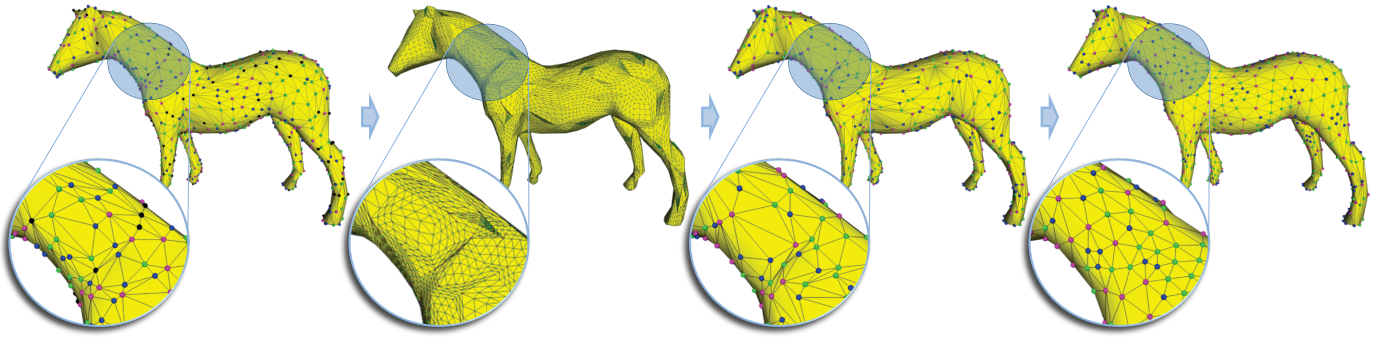

Figure 1: A triangle mesh (left) with low- and high-valence vertices (marked by black dots) remeshed into a 5-6-7 mesh with the same vertex count (right). Intermediate steps produce an initial 5-6-7 mesh (middle left) and a decimated version (middle right) which preserves the 5-6-7 property. The final 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.

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. An overview of our remeshing algorithm is shown in Figure 1.

2. Related Works

The quality of a surface mesh is crucial for a variety of applications such as 3D visualization and numerical simulation. Therefore, there has been an abundance of remeshing algorithms proposed in the literature to improve the quality of a surface mesh [3]. Some remeshing algorithms are based on improving the geometry of the mesh by redistributing the points on the underlying surface, e.g. [4] among many others. Other works look more into the mesh connectivity, e.g. [5, 6] 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. [7] 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 [8] 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.

Remeshing methods based on centroidal Voronoi tessellations (CVT) [9, 10, 11, 12] tend to generate meshes whose vertex valences are 6-dominant, as the majority of the Voronoi cells are hexagon when CVT converges. However, we are not aware of theoretical guarantees that the results would be completely void of Voronoi cells with < 5 sides or > 7 sides.

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. [13, 14, 15, 16] or use importance sampling [17] 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. [18, 4, 19] among others that can be found in the survey by Alliez et al. [3]. 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 by Li et al. [6], where their editing operations operate on a 5-6-7 mesh.

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

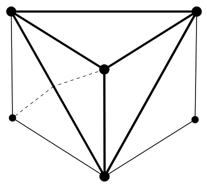

Figure 2: A simple scheme to remove a vertex of valence 3. A new vertex and two edges, dashed lines, are added to increase the degree of the v3 vertex by one.

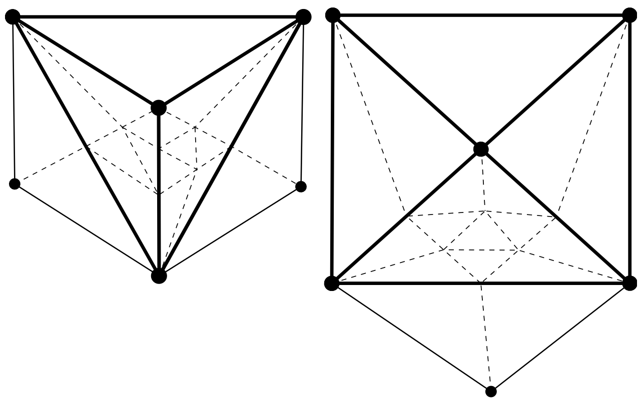

Figure 3: Elimination of a vertex with valence 3 (left) or 4 (right). Solid lines represent the edges in the original mesh and dashed lines are the new edges added by the remeshing process. Note that the original mesh edges are preserved and no geometry error is introduced.

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. Figure 4 depicts the 3D results of removing low valence vertices on a cube.

As an implementation detail, note that it is possible to have overlapping vertices, edges and faces, when eliminating low valence vertices in Figure 3. Overlapping occurs when two mesh elements (vertex, edge or face), which appear distinct in the subdivision scheme, happen to be the same element in the mesh geometry. While overlapping vertices and edges do not pose issues for our algorithm, handling overlapping faces can be tricky. It can only occur in the v3 removal scheme, with the two faces outside the one-ring neighborhood. Specifically, in a manifold configuration, it only happens in one pathological case, i.e. Tetrahedron. Note that a Tetrahedron has only four faces, while we have five faces drawn in our V3 removal scheme in Figure 3 and therefore, two of them are referring to the same face in the mesh. We work around this unique case by adding those two last edges sequentially rather than at the same time.

Also note that these structures only introduce new vertices that are $v567$ and may only 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 5.

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 6 illustrates the iterations involved for removing a vertex of valence 14.

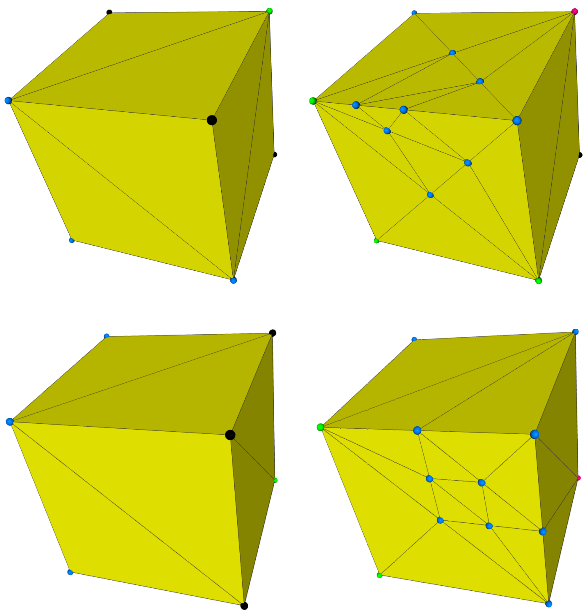

Figure 4: Demonstration of Valence 3(top) and Valence 4(bottom) removal on a cube.

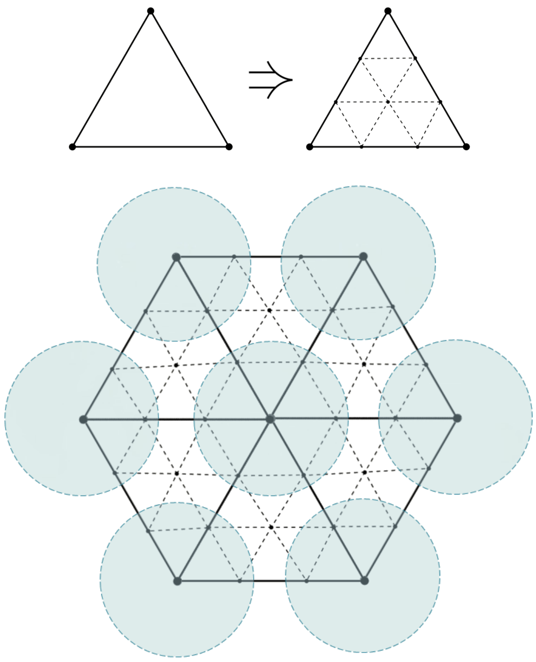

Figure 5: Topological subdivision to separate all the high-valence vertices sufficiently far apart from each other by at least two $v6$ vertices. The one-ring neighborhood that is not shared with other possible high valence vertices is indicated by the circles.

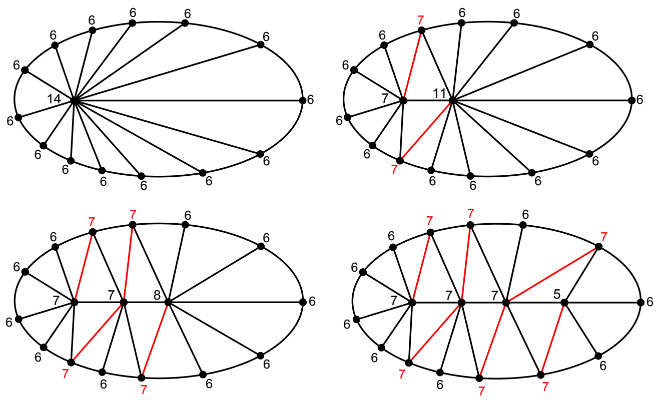

Figure 6: Elimination of a high-valence vertex ($v14$) by a series of vertex splits, where the ellipse denotes the one-ring neighbourhood of the vertex. This adds $\lfloor \frac{14-2}{3} \rfloor - 1 = 3$ of $v7$ vertices and one $v567$ vertex (in this example, the $v5$ vertex in the bottom-right image), while increasing the valence of the neighbour vertices by at most one (i.e. from $v6$ to $v7$).

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 7(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 $5 \leq d_1 + d_2 - 4 \leq 7$ 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.

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 7 (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.

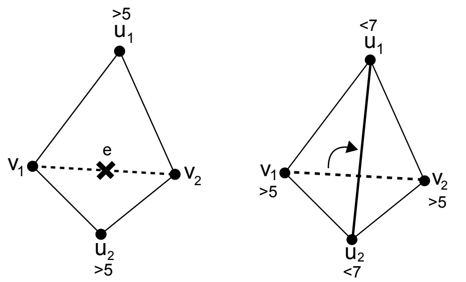

Figure 7: 5-6-7 preserving edge collapse (left) and edge flip (right).

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 8, 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.

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 enhance-

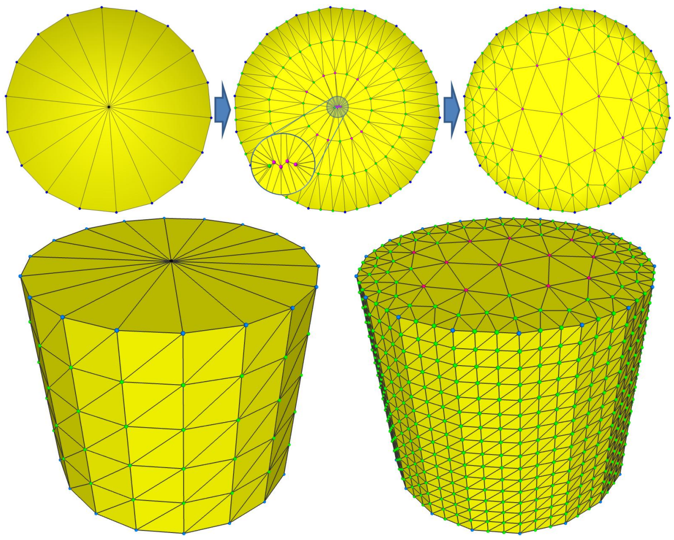

Figure 8: Comparison of the original cylinder mesh(left) with the 5-6-7 enhanced mesh(right). The top row from left to right: the original, 5-6-7, and 5-6-7 enhanced mesh (without decimation).

ment 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 8 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 and then geometically enhanced afterwards. The 5-6-7 remeshing has a linear time complexity, however the decimation process can be slow at and take minutes to run on a mesh with 25K vertices. Although quadric mesh decimation has a well defined time complexity, the running time of the constrained quadric decimation used in our work highly depends on the initial triangulation and how lucky we are in the progression of the quadric decimation. There are cases where the mesh can be decimated to the initial size, but in some cases it can take several alternations between edge collapses and edge flips to achieve the desired size.

5.1. Parameters

The first phase of our 5-6-7 remeshing does not require parameter tuning or user interaction to produce the 5-6-7 mesh, except for a parameter α that is used to control the distance of the split vertices from the original location of the high pivot vertex. Value of zero corresponds to geometric degeneracy and zero error, and higher values may add distortion in that locality to the mesh.

During the decimation step, we switch between edge collapses and edge flips, and edge flips can produce geometric error. Therefore, only edges with almost flat incident faces are allowed to be flipped. To detect the flatness, the dot product of the normal of the faces has to be below some threshold β, for that edge to be legit for a flip.

The geometry enhancement step needs two parameters: s and $\sigma_{threshold}$. Parameter s governs the speed of Laplacian smoothing. A value of 1.0 would be the basic Laplacian smoothing. And, parameter $\sigma_{threshold}$ is the rigidity factor described in the previous section, and is used in the feature detection step.

In our implementation, we used the values of $\alpha = 50$, $\beta = 0.1$, and $s = 0.1$ which are fixed for all of the experiments. Only the value of $\sigma_{threshold}$ needs to be tuned for each model, for which we provide an interactive tool for the user to pick an appropriate value for it.

5.2. Test data

The results of applying our algorithm on meshes with different topologies and feature types are shown in figure 13. The eight mesh shows the result of applying our method on a mesh of genus two and the other three exhibit our feature preservation mechanism on different feature types.

As shown in the second column of the meshes in Figure 13, in the vicinity of the low/high valence vertices, we have added clusters of vertices to remove them locally. In order not to have those dense clusters, we incorporate edge lengths in quadric errors such that, shorter edges become more suitable 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 [20], which calculates the Hausdorff distance between the original mesh and the final mesh after geometry enhancement. The error is mostly the result of the simplification process, as the remeshing phase 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.

5.3. Evaluation

In this section we would like to carry out a few more evaluations to show the advantages of 5-6-7 meshes. Although we compared some example meshes with their 5-6-7 version, one might still argue that this comparison is a bit biased because the initial mesh might have a very bad vertex positioning, and we are comparing it with a mesh that is geometrically enhanced at the end of the algorithm. So for a fair evaluation, we also provide comparison of our final 5-6-7 mesh with the geometrically enhanced version of the initial mesh in Figure 10.

It can be seen that, the existence of irregular vertices, especially in the flat regions of the mesh, enforces a non-uniform distribution of the points even after applying the Laplacian relaxation on the surface of the mesh. Although this relaxation tries to improve the point distribution on the surface of the

Model	Error	# Faces	Valence					Time	Compression (bpv)	
			V_{low}	V_5	V_6	V_7	V_{high}		Original	5-6-7
fish	0.010	1K	8.9%	24.5%	36.1%	22.5%	8%	3.3s	2.36	1.69
horse	0.018	1.5K	7.7%	27.7%	33.6%	22.1%	8.9%	4.5s	2.38	1.66
venus	0.020	1.5K	9%	25.7%	35.6%	19%	10.7%	4.2s	2.35	1.36
eight	0.010	1.5K	5.1%	23.2%	42.6%	23.9%	5.2%	5.1s	2.10	1.65
cow	0.020	6K	3.2%	17.9%	61.7%	12.5%	4.7%	18.7s	1.77	1.67
armhand	0.007	25K	3.6%	26.2%	42.7%	22.2%	5.3%	13m	1.97	1.56

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.

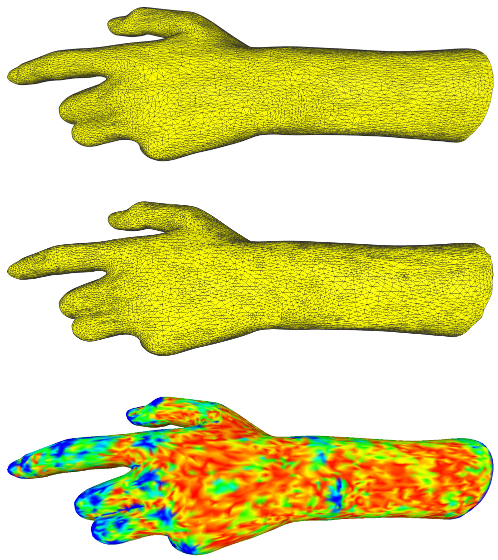

Figure 9: Visualization of Hausdorff distance error. Figures from top to bottom: the original mesh, the 5-6-7 mesh, the error visualization on the 5-6-7 mesh.

mesh, the non-uniform distribution of the points is inherent in the bad quality of the connectivity graph. For example, the effect is visible in the the v3 vertices near the eye of the fish, or the high valence vertex on the body of the fish.

Our next evaluation is to compare the result of subdivision algorithms on the original and 5-6-7 meshes. Figure 11 shows the results of applying two iterations of Loop subdivision [21] on an initial and the 5-6-7 remeshed sphere. Figure 12 presents the same results, with material, lighting and under different shading algorithms. Wrinkles appear in the smooth shaded version of the original mesh after applying subdivision, which is an inevitable artifact of subdivision algorithms around high valence vertices. Although in our 5-6-7 mesh we have more non-valence-6 vertices(rather than two extremely high valence vertices) but since their valences are bounded well enough, the artifact is not apparent.

Compression: In addition, 5-6-7 meshes result in noticeable improvement in connectivity compression. We utilized the benchmark tool provided by Isenburg et al. [22] and as the results are shown in Table 1, on 5-6-7 meshes we achieve an enhancement of about 1.5X on average. However, 5-6-7 meshes might not be the best choice for connectivity compression, since a v6 dominant mesh with few high/low valence vertices might be more suitable for comparing to a 5-6-7 mesh with almost equal number of v5, v6 and v7 vertices. The values shown in the table are the average bit per vertices, which is calculated as the total amount of bits required to encode the mesh divided by the number of vertices.

Finally, in order to verify the integrity of our feature preservation, we plot the per vertex error between the initial mesh and the 5-6-7 mesh. The color of each vertex corresponds to its Hausdorff distance from the original mesh, and the error increases as we go from blue to red. As it is shown in Figure 9, the feature areas of the hand are relatively preserved.

5.4. Limitations

Time Complexity: One of the drawbacks of our remeshing algorithm is the time complexity of the decimation step. Although other parts of the algorithm have a linear time complexity and run efficiently, the decimation step requires alternating between the edge collapses and edge flips, till it reaches the original mesh size. In the worst case the time complexity would be $O(E^2 logE)$ in total. However, in most of the experiments, we observed that at each edge collapse iteration, the size of the mesh is reduced to half. Which brings it down to $O(ElogE)$.

Subdivision Step: The subdivision step multiplies the size of the mesh by 9, which is not desirable for most of the large meshes. However, we were not able to find an alternative to push irregularities away from each other.

Geometry Enhancement Step: The geometry enhancement step needs some human interaction to pick the proper rigidity threshold. Parameter tuning on one hand, and the possible face fold overs on the other hand are the main downsides of this step. Face fold overs happen mainly because our feature preservation fixes feature points, allowing other vertices to move. One possible improvement would be allowing feature vertices to move along the feature creases.

6. Analysis and discussion

In this section, we present some observations about 5-6-7 meshes and discuss some theoretical aspects of this class of meshes. We show an interesting observation relating the number of vertices and the genus of the shape, which is useful in

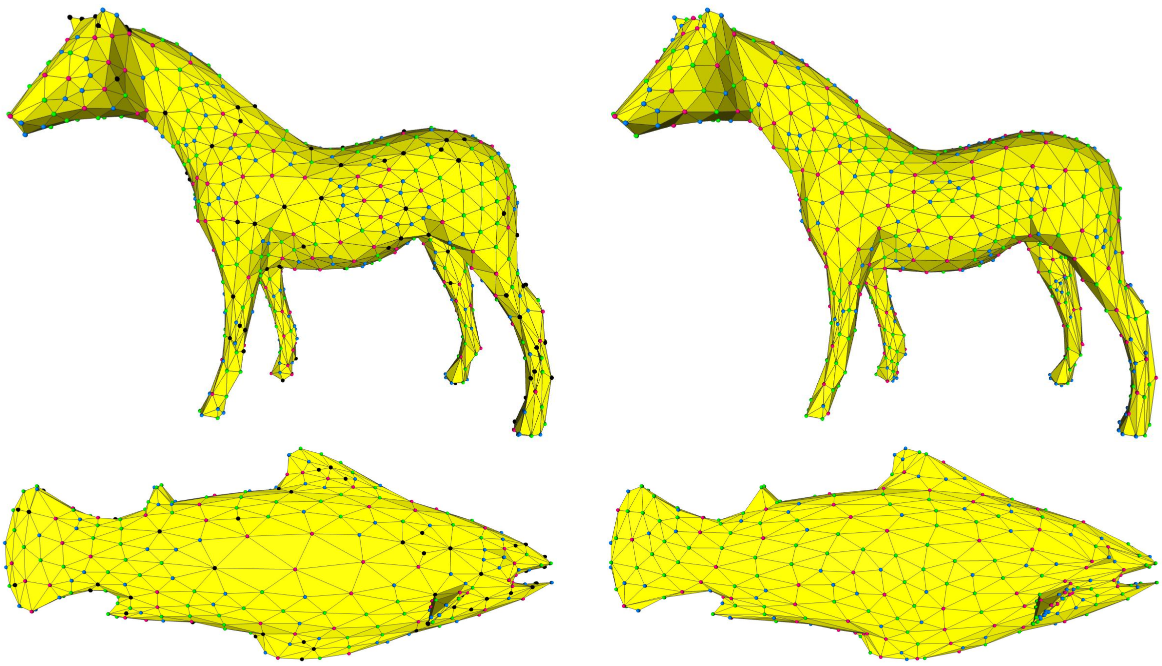

Figure 10: Comparison between the geometrically enhanced version of the initial mesh and the 5-6-7 mesh. On the left you see the initial mesh after running the geometry enhancement algorithm on it, versus the 5-6-7 mesh on the right.

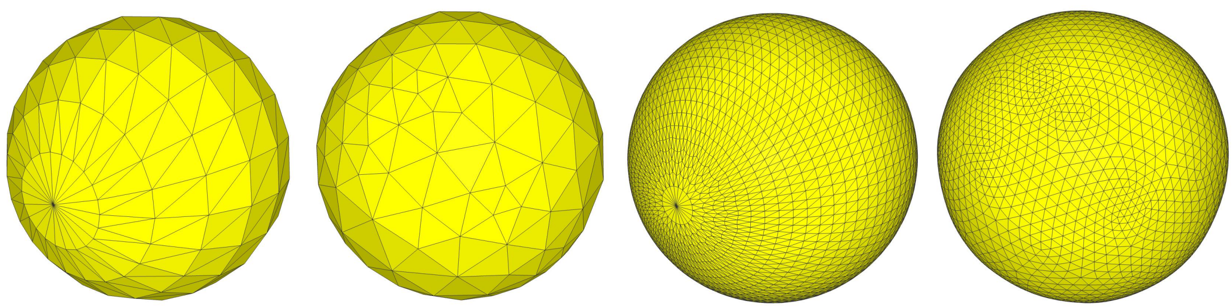

Figure 11: Results of subdivision on a sphere and the 5-6-7 sphere. Figures from left to right: 1) initial sphere 2) 5-6-7 sphere with the same vertex count 3) initial sphere after applying 2 steps of Loop subdivision 4) 5-6-7 sphere after applying 2 steps of Loop subdivision.

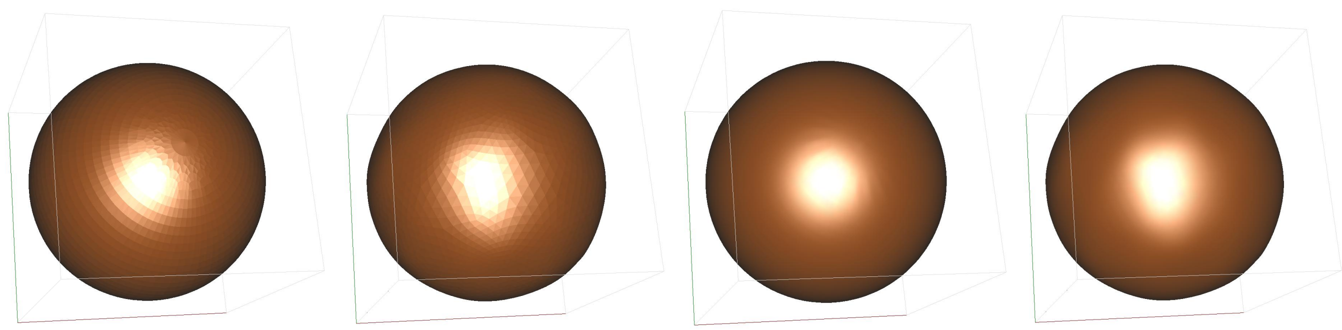

Figure 12: Shaded results of subdivision on a sphere and the 5-6-7 sphere. Figures from left to right: 1) initial sphere flat shaded 2) 5-6-7 sphere flat shaded 3) initial sphere smooth shaded 4) 5-6-7 sphere smooth shaded

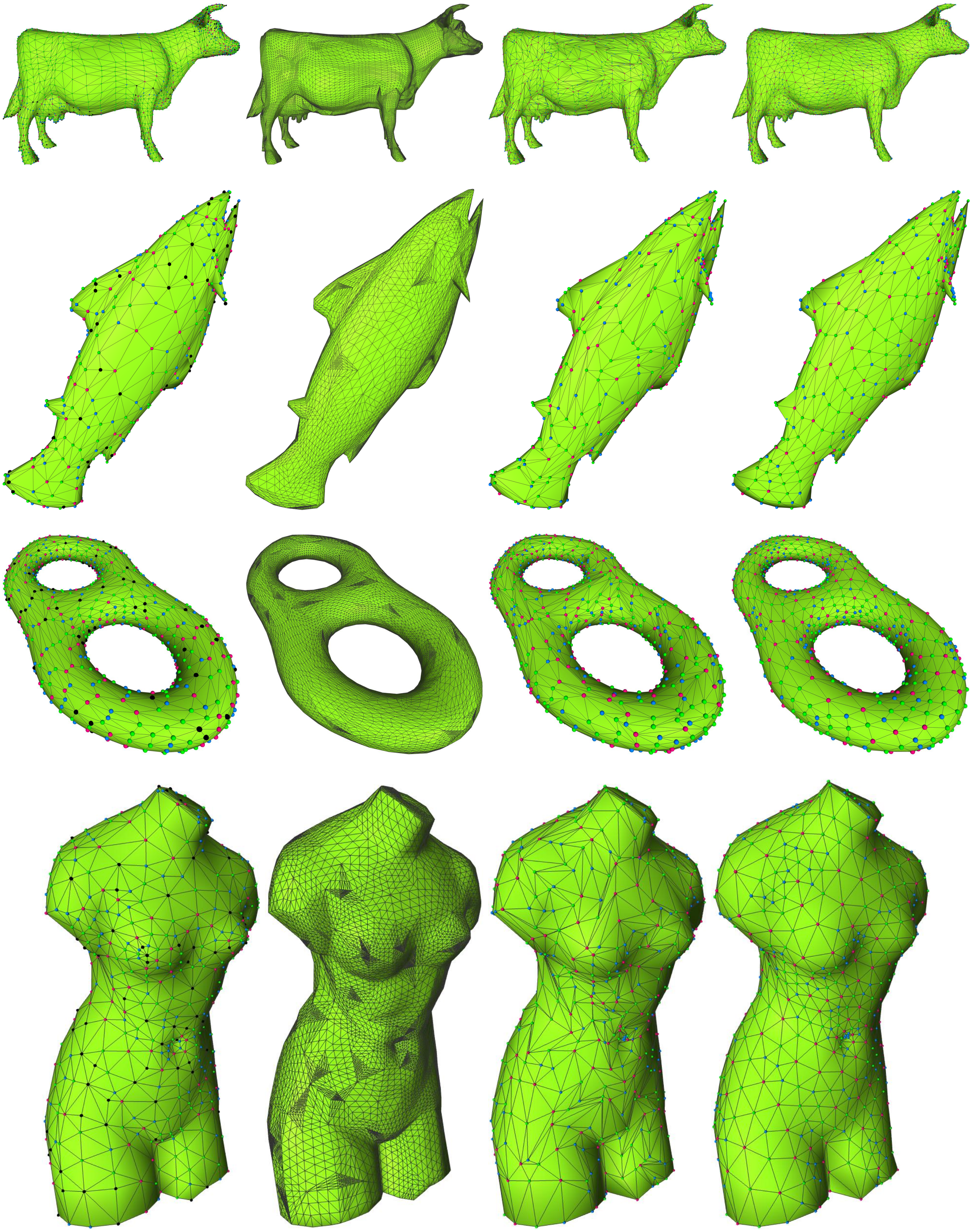

Figure 13: An example of feature preservation on a complex mesh(1st row), a mesh with fairly flat areas as well as some sharp feature regions(2nd row), a mesh with genus 2(3rd row), and the mesh of venus with subtle feature areas (4th row). From left to right: the original mesh, the 5-6-7 mesh, the decimated 5-6-7 mesh and finally the enhanced 5-6-7 mesh.

designing "minimal" 5-6-7 meshes. We then talk about alternative approaches to remove non-5-6-7 vertices.

6.1. Count of 5-6-7 vertices

In this section, we study the relation between the number of v5, v6, and v7 vertices on a mesh of an arbitrary genus. The results of this study can be beneficial in designing "minimal" 5-6-7 meshes and having a better understanding of the relationship between irregularities and the genus of the shape in an abstract sense. We start by studying the number of v5 and v6 vertices in a 5-6 mesh, then we extend it to 6-7 meshes and finally, we generalize it to 5-6-7 meshes. Knowing that the average valence of vertices on a mesh with genus zero is less than 6, we would like to examine the possibility of having a 5-6 remeshing. It is also interesting to see the relation between the number of v5 and v6 vertices. Let n_5, n_6, and n_7 be the number of v5, v6, and v7 vertices, and g be the genus of the mesh. Also let V, E, and F denote the number of vertices, edges, and faces of the triangle mesh. we have:

$$V = n_5 + n_6$$

$$E = \frac{5n_5 + 6n_6}{2}$$

$$F = \frac{5n_5 + 6n_6}{3}$$

Using Euler's characteristic formula ($V - E + F = 2 - 2g$), we get:

$$n_5 = 12\,(1 - g) \tag{1}$$

It is interesting that a 5-6 remeshing of a mesh with genus zero, requires exactly twelve v5 vertices, and any number of v6 vertices. Therefore, a mesh with minimum number of vertices on genus zero would have no v6 vertices and exactly twelve v5 vertices, which is an Icosahedron. Following a similar calculation, for a 6-7 remeshing of meshes with genus higher than one, we get:

$$n_7 = 12\,(g - 1) \tag{2}$$

Intuitively it means that an increase in the genus corresponds to an addition of twelve v7 vertices to the mesh, independently from the number of v6 vertices. Generalizing the approach for 5-6-7 meshes, we get:

$$V = n_5 + n_6 + n_7$$

$$E = \frac{5n_5 + 6n_6 + 7n_7}{2}$$

$$F = \frac{5n_5 + 6n_6 + 7n_7}{3}$$

$$(n_5 + n_6 + n_7) - \frac{5n_5 + 6n_6 + 7n_7}{2} + \frac{5n_5 + 6n_6 + 7n_7}{3} = 2 - 2g$$

$$n5 - n7 = 12(1 - g) \tag{3}$$

The useful observations are:

- An exact relation between the number of the irregularities and the genus.
- An addition of every extra v5 vertex, needs to be compensated with a v7 vertex, and vice versa.

These observations hold for 5-6-7 triangle meshes and also for any 5-6-7 remeshing that preserves the genus of the shape, which suggests that a minimal local remeshing of a low valence vertex should contain only v5 vertices. Since added v7 vertices should be compensated with v5 vertices, and adding v6 vertices is arbitrary, an optimal remeshing should only contain v5 vertices. Nonetheless, it is not true that every local remeshing scheme with only v5 vertices is minimal, due to the added valencies in the locality.

It should be noted that even though Euler's characteristic formula can give a theoretical lower bound for 5-6-7 meshes, there are cases that the minimal 5-6-7 mesh suggested by Euler's formula is degenerate (E.g $V = 0$ and $g = 1$). Moreover, in other cases the resulting 5-6-7 mesh cannot have planar triangles and straight edges ($g > 1$). This is because Euler's formula does not have any assumptions on the geometric characteristics of the graphs, and only suggests an abstract connectivity. This abstract connectivity may suite our purposes or may require us to add extra v6 vertices to produce triangle meshes with planar faces and straight edges.

6.2. Alternative structures to remove high/low valence vertices

There are several possible schemes to remove high/low valence vertices. Our choice of the high valence removal structure is motivated by adding a relatively low number of new vertices, while respecting the geometry. To remove high valence vertices, we present two additional possible schemes, shown in Figure 14. The first scheme, shown on the top, has a regular structure and can be suitable for cases that we require a better control over the vertices of the mesh to reduce the geometric error, or in cases that we require a better distribution of the new points. In this structure, for a vertex of degree $3d$, $\frac{d(d-1)}{2}$ new regular vertices are added. Otherwise, if the initial degree is not a multiple of three, we can insert one or two new vertices anywhere on the boundary and convert one or two regular vertices to a v7 vertex. Note that in this case, the degree of all of the vertices on the boundry, except for 6 of them, is increased by one.

The second scheme, shown at the bottom, mimics a rotationally symmetric structure and can be suitable for geometric cases such as the poles of a sphere. In this structure, we have a set of nested rings, and moving towards the center, the number of irregular vertices in each ring decreases by one. Most of the vertices are regular and the total number of added vertices for a vertex of degree d is $\frac{d(d-1)}{2} - 21$.

One can also remove a high valence vertex, and triangulate its one-ring neighborhood by zigzagging on the boundary, which increases the valence of all the vertices on the boundary except for two of them. The downside is that, this approach will completely discard the geometry of the original vertex.

Alternative schemes for low valency removal, are shown in Figure 15 for V4 removal and in Figure 16 for V3 removal. These new structures produce less vertices than the previous schemes and similarly preserve the shape of the mesh and do

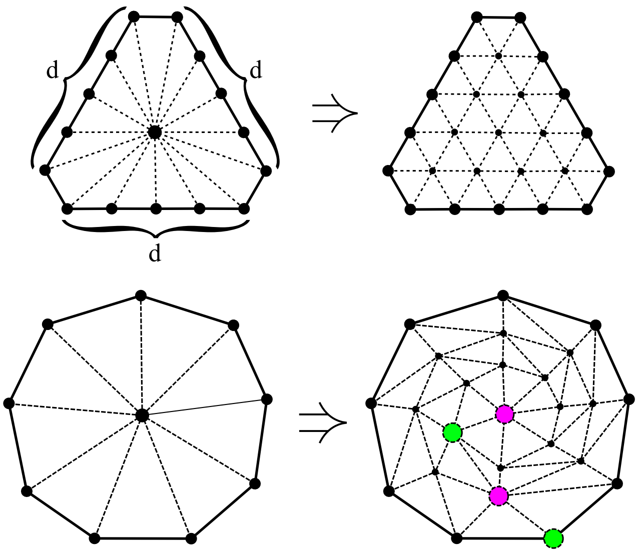

Figure 14: Two alternative schemes to remove a high valence vertex.

not add geometric error after being applied. However, we used the structures in Figure 3 since they modify a smaller region around the low valence vertex.

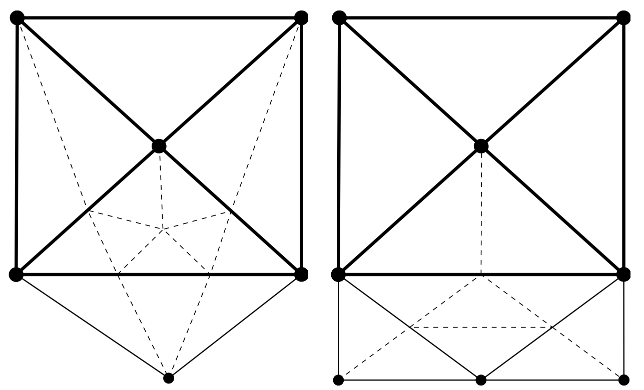

Figure 15: Two alternative schemes to remove a vertex of degree 4.

7. Conclusion and Future work

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).

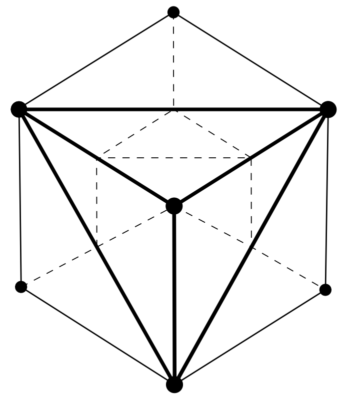

Figure 16: An alternative scheme to remove a vertex of degree 3.

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.

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, an interesting question is the placement of valence 5 and 7 vertices for which it may be desirable to consider the density of the points and the frequency of the mesh.

As we mentioned in the analysis section, Euler characteristic formula does not take into account any information about the geometry of the shape. But in most of the applications in computer graphics, we require the triangles to be planar and the edges to be straight lines. Therefore, an alternative formulation to incorporate some constraints from the geometry into the connectivity would be a valuable work to find minimal 5-6-7 meshes with planar triangles and straight edges.

Acknowledgement

We first thank the anonymous reviewers from both Graphics Interface 2012 and this journal for their valuable comments. Thanks should also go to the members of the Graphics Usability and Visualization (GrUVi) lab at Simon Fraser University (SFU) 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).

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. ☐

References

[1] Luebke D, Watson B, Cohen JD, Reddy M, Varshney A. Level of Detail for 3D Graphics. Elsevier Science Inc.; 2002.

[2] Peters J, Wu X. The distance of a subdivision surface to its control polyhedron. J Approx Theory 2009;:491–507.

[3] Alliez P, Ucelli G, Gotsman C, Attene M. Recent advances in remeshing of surfaces. In: Shape Analysis and Structuring. Springer Berlin Heidelberg; 2008, p. 53–82.

[4] Surazhsky V, Gotsman C. Explicit surface remeshing. In: Proc. Symp. on Geom. Processing (SGP). 2003, p. 20–30.

[5] han Peng C, Zhang E, Kobayashi Y, Wonka P. Connectivity editing for quadrilateral meshes. ACM Trans on Graphics (Proc SIGGRAPH Asia) 2011;.

[6] Li Y, Zhang E, Kobayashi Y, Wonka P. Editing operations for irregular vertices in triangle meshes. In: ACM Trans. on Graphics (Proc. SIGGRAPH Asia). 2010, p. 153–65.

[7] Isenburg M, Gumhold S, Gotsman C, Gumhold S, Gotsman C. Connectivity shapes. In: In IEEE Visualization 2001 Conference Proceedings (2001. 2001, p. 135–42.

[8] Velho L. Using semi-regular 4-8 meshes for subdivision surfaces. J Graph Tools 2000;:35–47.

[9] Du Q, Gunzburger MD, Ju L. Constrained centroidal voronoi tessellations for surfaces. SIAM Journal on Scientific Computing 2002;24(5):1488–506.

[10] Liu Y, Wang W, Lévy B, Sun F, Yan DM, Lu L, et al. On centroidal voronoi tessellation–energy smoothness and fast computation. ACM Trans on Graphics 2009;28(4):1–17.

[11] Rong G, Jin M, Guo X. Hyperbolic centroidal voronoi tessellation. In: Proceedings of the 14th ACM Symposium on Solid and Physical Modeling (SPM 2010). 2010, p. 117–26.

[12] Rong G, Jin M, Shuai L, Guo X. Centroidal voronoi tessellation in universal covering space of manifold surfaces. Computer Aided Geometric Design 2011;28(8):475–96.

[13] Chiang CH, Jong BS, Lin TW. A robust feature-preserving semi-regular remeshing method for triangular meshes. The Visual Computer 2011;27(9):811–25.

[14] Vidal V, Wolf C, Dupont F. Combinatorial mesh optimization. The Visual Computer 2012;28(5):511–25.

[15] Hormann K, Labsik U, Greiner G. Remeshing triangulated surfaces with optimal parameterizations. Computer-Aided Design 2001;:779–88.

[16] Alliez P, Meyer M, Desbrun M. Interactive geometry remeshing. ACM Trans Graph 2002;:347–54.

[17] Corsini M, Cignoni P, Scopigno R. Efficient and flexible sampling with blue noise properties of triangular meshes. IEEE Transactions on Visualization and Computer Graphics 2012;18(6):914–24.

[18] Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Mesh optimization. In: Proceedings of the 20th annual conference on Computer graphics and interactive techniques. SIGGRAPH '93; 1993, p. 19–26.

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

[20] Cignoni P, Rocchini C, Scopigno R. Metro: Measuring error on simplified surfaces. Computer Graphics Forum 1998;:167–74.

[21] Loop C. Smooth Subdivision Surfaces Based on Triangles. Department of mathematics; University of Utah; 1987.

[22] Isenburg M, Lindstrom P, Snoeyink J. A benchmark coder for polygon mesh compression. 2002. URL `http://www.cs.unc.edu/isenburg/pmc/`.