Editing Operations for Irregular Vertices in Triangle Meshes

Yuanyuan Li
Arizona State University

Eugene Zhang
Oregon State University

Yoshihiro Kobayashi
Arizona State University

Peter Wonka
Arizona State University

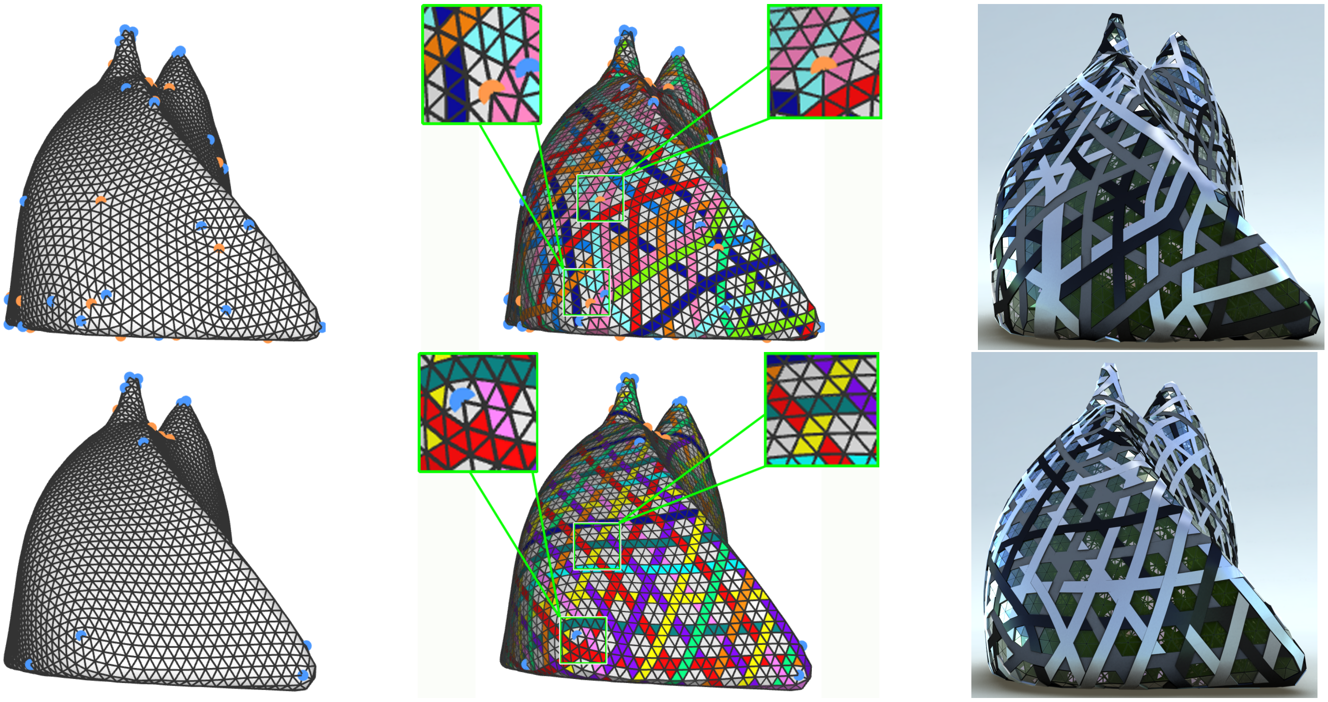


Figure 1: *Our research question is what edits are possible to control the irregular vertices of a triangle mesh. We visualize the impact of irregular vertices in a design application. Top: A designer-generated mesh (left) is used as input to assign panel ids for a procedural geometry replacement system (middle and right). The irregular vertices force strips of width two as well as many degenerate shapes between strips. Bottom: Editing the type, location, and number of irregular vertices (left) yields a design with smooth strips of width one (middle and right).*

Abstract

We describe an interactive editing framework that provides control over the type, location, and number of irregular vertices in a triangle mesh. We first provide a theoretical analysis to identify the simplest possible operations for editing irregular vertices and then introduce a hierarchy of editing operations to control the type, location, and number of irregular vertices. We demonstrate the power of our editing framework with an example application in pattern design on surfaces.

Keywords: triangle mesh editing, topology, irregular vertices

1 Introduction

In this paper we introduce mesh editing operations to control the type, location, and number of the irregular vertices in a triangle mesh. An irregular vertex is a vertex that has a valence not equal to six. The existence of irregular vertices is linked to the topology and geometry of the underlying surface. Topologically, given a closed mesh surface whose *Euler characteristic* is not zero, there is at least one irregular vertex. Geometrically, irregular vertices naturally appear in high discrete Gauss curvature regions in the surface, with their types dependent on the Gauss curvature. The ability to control the type, location, and number of irregular vertices in a mesh is important in several applications such as isotropic remeshing, pattern synthesis, architectural design, mesh-based modeling, simplification, and subdivision surfaces. For example, in isotropic remeshing irregular vertices in flat regions can lead to angular distortions. The appearance of irregular vertices can also cause visual artifacts in circle packing for architectural designs and interpolation difficulties for subdivision surfaces. Our work provides what we believe to be the first direct control mechanism for the irregular vertices in a triangle mesh.

Controlling the irregular vertices in a triangle mesh is a challenging problem. Previous work in mesh optimization mainly focuses on preserving the geometric details and the shape of the triangles, while the topological properties of the triangle mesh are not treated very systematically. Some work [Hoppe et al. 1993; Surazhsky and Gotsman 2003] mentions topological editing, but only very briefly.

ACM Reference Format
Li, Y., Zhang, E., Kobayashi, Y., Wonka, P. 2010. Editing Operations for Irregular Vertices in Triangle Meshes. *ACM Trans. Graph. 29*, 6, Article 153 (December 2010), 11 pages. DOI = 10.1145/1866158.1866179 http://doi.acm.org/10.1145/1866158.1866179.

The challenge of this work is that these discrete topological operations are not very intuitive. For example, when designing continuous vector and tensor fields we can draw from Poincaré and Conley theories to move a singularity and cancel a singularity pair. Therefore, it is somewhat surprising that in the discrete case it is not possible to move a single irregular vertex or to cancel a valence 5 and a valence 7 irregular vertex pair. We therefore argue that previous work did not even establish what operations are possible and how these possible operations could be performed. Unlike vector and tensor field editing [Zhang et al. 2006; Zhang et al. 2007; Palacios and Zhang 2007; Fisher et al. 2007; Ray et al. 2008], the atomic operations involving irregular vertices require at least two irregular vertices for movement and three for cancellation. Consequently, defining the functionalities of irregular vertex editing, both atomic and composite, is a needed task for which no prior solution has been given. The second difficulty is related to the implementation of these operations. For example, what lower-level operations are needed in order to implement an atomic irregular vertex editing operation? Also, given a higher-level editing operation, how can it be realized through atomic editing operations. Besides the aforementioned applications, the main motivation behind our work is intellectual curiosity. What is possible and impossible when it comes to controlling the irregular vertices in a triangle mesh? The main idea of this paper is to use the three basic topological operations, *edge flip*, *edge collapse*, and *vertex split* to redistribute the valence deficit over the mesh (hence move the irregular vertices), such that the type, location, and number of the irregular vertices can be edited interactively. The contributions of this paper are:

- We analyze what atomic operations are possible on triangle meshes for the movement, cancellation, clustering, merging, split, and generation of irregular vertices.
- We provide efficient implementation of these editing operations based on well known low-level graph editing operations.
- We develop a framework to interactively edit the irregular vertices in a triangle mesh. Composite editing operations are identified and their implementation based on atomic editing operations is provided.

We demonstrate the power of our system with regular texture and geometry synthesis on surfaces, e.g. see Fig. 1.

2 Related Work

We review related work in mesh optimization and field editing.

Mesh Optimization: Our work can be considered as a complement to many existing mesh optimization algorithms discussed in the literature, e.g. mesh smoothing [Desbrun et al. 1999]. To construct higher-level editing operations we combine low-level editing operations discussed in the literature, such as *edge collapse* [Hoppe 1996], vertex split, and edge flip. An alternative set of low-level operations can be derived from Stellar theory [Lewiner et al. 2010]. Our work shares some common goals with triangular remeshing [Turk 1992; Gu et al. 2002; Alliez et al. 2002; Alliez et al. 2003; Surazhsky et al. 2003], but the control of irregular vertices is not treated systematically in previous work. An interesting exception is the work by Surazhsky and Gotsman [2003]. They also suggest to use edits based on *edge flips* to reduce the number of irregular vertices, but their proposed work does not have enough editing operations and only few operations are possible. Akleman and Chen argue why extremal points and saddles should be modeled with irregular vertices [2006]. The location of irregular vertices influences inverse subdivision algorithms [Taubin 2002]. Note that our goal is different from topological mesh processing and shape editing [Nealen et al. 2005]. These approaches are also complementary to our work.

Vector and Tensor Field Design: Irregular vertices in a mesh play a similar role of singularities in a vector field or umbilical points in the curvature tensor. They all represent some irregularity in their respective domain. It is therefore not surprising we borrow notions from vector and tensor field design for the control over irregular vertices [Zhang et al. 2006; Zhang et al. 2007; Palacios and Zhang 2007; Fisher et al. 2007; Ray et al. 2008; Bommes et al. 2009]. Due to the index theory, a pair of singularities with opposite indexes must be removed simultaneously [Zhang et al. 2006]. On the other hand, a singularity can be moved without impacting the topology of the field. In fact, singularity pair cancellation and movement are the atomic operations to control singularities in vector and tensor fields, and their implementation is based on Poincaré and Conley theories. However, we have found that it is impossible to move an irregular vertex without introducing more irregular vertices and to cancel an irregular vertex pair with opposite discrete Gauss curvatures.

3 Overview

The input to our system is a triangle mesh M that represents a closed manifold surface. The *valence* of a vertex v in M, which we denote as $l(v)$, is the number of edges in the mesh incident to v. A vertex with a valence of n is denoted as vn, e.g., v5 and v7. We set out to define effective and comprehensive semantic editing operations to control the type, location, and number of irregular vertices. We structure these desired editing operations in four groups: altering the valence of irregular vertices (type change), changing the location of irregular vertices (move), decreasing the number of irregular vertices (remove), and increasing the number of irregular vertices (generation). To avoid a convoluted exposition we exclude irregular vertices with valence that are multiples of 6 from the remainder of the overview section.

Type change: The most fundamental and thus common irregular vertices are v5 and v7. Any other irregular vertex can be converted into multiple close-by v5 or v7 vertices. For example, a v4 vertex can be split into two v5 vertices connected by two edges. The other direction is not as simple as there are some constraints on the arrangement, e.g. two adjacent v5 vertices cannot be converted into a v4 vertex. However, other operations (moving, generating, and removing) of vertices with valence other than 5 and 7 can be performed by converting to v5 and v7 before the editing operation and then converting back afterwards. This is because the vertices typically stay in an appropriate configuration that can be converted back. In our implementation of the other operations we therefore focus on v5 and v7 vertices and provide the user with type change operations to manipulate irregular vertices of valence other than 5 and 7. See Section 6.3 for type change operations.

Move: Moving a set of irregular vertices refers to changing the location of the vertices while the other irregular vertices are not impacted. Analogous to previous work in field design, we would have liked to design an interface that enables the user to select a single irregular vertex and move it using drag and drop. Unfortunately, this is impossible for vertices whose valence is not a multiple of 6. We provide an analysis of the problem in Section 5. This is somewhat surprising and makes irregular vertex control a challenging problem. The most fundamental operations that we provide thus are for the movement of irregular vertex pairs. In all cases the user can select two vertices and specify the movement direction of one of the vertices. The movement direction of the other vertex is always specified implicitly and has no more degree of freedom. The three most important operations are the following: 1) moving a $5-7$ vertex pair. Both vertices can move over the mesh similar to translation or parallel transport [Zhang et al. 2006]. The relative distance and relative orientation between the vertex pair stay the same. 2) moving a $5-5$ vertex pair. The vertices can move closer together, further apart, or circle around a fixed point in between the two vertices. This is only an intuitive description and the precise results will be given in Section 7.1. 3) moving a $7-7$ pair, which has a behavior similar to moving a $5-5$ pair.

Remove: From Poincaré index theory and Conley index theory we know that removing a single irregular vertex is not feasible. However, surprisingly canceling two irregular vertices is also not possible (see Section 5). This is another reason why valence control is difficult. While it is possible to cancel four irregular vertices, they have to be in a specific and uncommon configuration. The two most important removal operations operate on irregular vertex triples. A $5-7-7$ triple can be removed while generating a new v7 vertex. A $5-5-7$ triple can be removed while generating a new v5 vertex. Removal operations are described in Section 7.2.

Generation: Generation of irregular vertices is the inverse operation of removal. We can therefore use the same mechanisms as removal. The only practical difference is that generating a quadruple from a regular vertex is always possible, while finding a quadruple in the right configuration on an existing mesh is difficult. See Section 7.3.

In addition to the aforementioned topological editing operations, geometric operations to optimize the shape of triangles are also provided (see Section 8). We structure our semantic editing operations into fundamental semantic editing operations and composite semantic editing operations. Before we can explain the most powerful operations described above, we will lay the groundwork with fundamental semantic editing operations in Section 6.

4 Background

In this section we briefly review most relevant results from algebraic topology and discrete differential geometry. Further, we present some of our own definitions and observations.

Given an ideal triangulation in which all triangles are equilateral, the valence of a vertex v is related to the discrete Gauss curvature of v, defined in the following fashion:

$$G(v) = (6-l(v))\frac{\pi}{3} \tag{1}$$

Consequently, $v6$ vertices are considered as *regular vertices* and all other vertices as *irregular vertices*. However, note that the definitions of regular and irregular vertices are purely topological, i.e., they do not require the ideal triangulation.

In the remainder of the paper we will often consider paths and regions on a mesh, which require the following definitions. A *path* γ (Fig. 2 left) on the mesh M consists of a sequence of edges $e_i = (v_i, v_{i+1})$ for $0 \leq i < N$. N is the *length* of γ. A path is a *loop* if $v_0 = v_N$. Otherwise, γ is an *open path*. A loop γ is *degenerate* if there exists a vertex in γ that is incident to at least three edges in γ. A *degenerate open path* can be defined in a similar fashion. In the remainder of the paper we will only consider non-degenerate open paths and loops.

A path (open path or loop) γ consisting of only regular vertices is *regular*. For a regular path γ, the edges in γ divide the 1-ring neighborhood of any interior vertex v on γ into two subsets of triangles. v is a *turn point* if the numbers of triangles on the two sides are different from each other. Otherwise, it is a *non-turn point*. A path γ is *straight* if every interior vertex in γ is regular and non-turning.

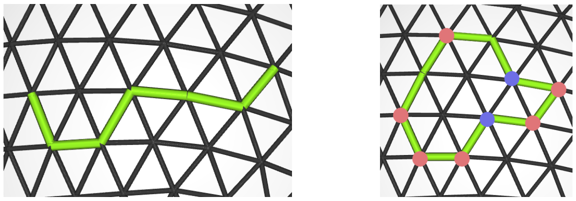


Figure 2: *Left: an open path. Right: a region whose boundary loop has turn points: convex (red) and concave (blue).*

A *region* R is a subset of the triangles in M whose dual graph is connected. That is, for any two triangles s and t in R, there is a sequence of triangles such that $t_0 = s$, $t_N = t$, and t_i and t_{i+1} share an edge in R for all $0 \leq i < N$. The boundary of R, denoted by ∂R, is a loop in non-degenerate cases. We further require that the boundary loop be regular, i.e., it does not contain irregular vertices. In this paper we are interested in properties of loops that can be derived from knowing only the exterior $\bar{R}$ (complement of R in the mesh). A turn point on ∂R is *convex* if it has more than 3 adjacent triangles in $\bar{R}$. If less than three triangles are in the exterior the turn point is *concave* (See Fig. 2). The *angle* of a turn v, denoted by $k(v)$, is $(m(v)-3)\frac{\pi}{3}$ where $m(v)$ is the number of incident triangles of v outside R. The angle of turn is positive for a convex turn point, negative for a concave point, and zero for a non-turn point. A region R is *convex* if there is no concave turn points on ∂R. Otherwise, it is *concave*.

The discrete Gauss-Bonnet theorem relates the total turning angle $\sum_{v\in\partial R} k(v)$ along the boundary ∂R to the total discrete Gauss curvature $l(v)$ of the interior vertices of R as follows:

$$\sum_{v\in\partial R}(m(v)-3)+\sum_{v\in intR}(6-l(v)) = 6\chi(M) \tag{2}$$

where $\chi(M)$ is the *Euler characteristic* of M.

Several results in this paper are derived by analyzing convex regions and their boundaries. The boundary loop of region R can be interpreted as a polygon. The turn points partition the loop into sides s_i. The number of edges of side s_i is denoted as b_i. If we consider a vertex v_0 in R then d_i is the graph distance of s_i to v_0 (See Fig. 4 left). For convenience, we will always introduce a zero-length side anytime an angle of turn is $\frac{2\pi}{3}$. This zero-length side is delimited by two virtual turn points each with a turn angle of $\frac{\pi}{3}$. For example, in Fig. 4 left b_5 equals 0 but we still consider the boundary loop as pentagon. We do not need to consider cases where the angle of turn is greater than $\frac{2\pi}{3}$ as such cases are degenerate. Consequently, there cannot be any consecutive zero-length segments.

Using Equation 2 and the consideration of zero-length sides and virtual turn points, it is easy to establish the following facts about a convex region R: 1) All turn points on the boundary of R have an angle of turn of $\frac{\pi}{3}$. 2) If R contains exactly one irregular vertex v_0 in its interior then the boundary of R has $l(v_0)$ turns. 3) If R contains no irregular vertex then the boundary of R is a hexagon, i.e. has 6 turns. 4) If R contains two irregular vertices v_1 and v_2 in its interior with $l(v_1)+l(v_2) = 12$ then the boundary of R is also a hexagon.

5 Impossible Editing Operations

The goal of our system is to provide the ability to control the type, location, and number of the irregular vertices without degrading the quality of the mesh. Similar goals have been set in vector and tensor field processing [Zhang et al. 2006; Zhang et al. 2007; Palacios and Zhang 2007; Fisher et al. 2007; Ray et al. 2008; Bommes et al. 2009] to control the number and location of singularities. Two types of fundamental operations have been proposed: singularity

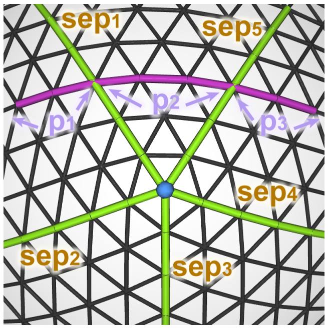


Figure 3: *A v5 vertex in blue and its five separatrices in green. The straight path in purple is divided into three pure segments p_1, p_2, and p_3. Only p_2 is distance preserving and p_1 and p_3 are distance varying.*

movement in which a singularity is moved to a more desirable location, and singularity pair cancellation, in which a singularity pair with opposite singularity indices is removed simultaneously from the field.

Unfortunately, the corresponding operations in our setting, i.e., the movement of an irregular vertex or the cancellation of an irregular vertex pair are in general impossible in the following sense:

Theorem 5.1 *Consider a convex region R that contains exactly one irregular vertex v_0 in its interior. If $l(v_0)$, the valence of v_0, is not a multiple of 6, then it is impossible to remesh the interior triangles of R to have a different configuration that still contains only one irregular vertex.*

To prove this statement we need the following:

Definition 5.2 *Given a mesh M and an irregular vertex v_0 with a valence of $l(v_0)$, a* valid neighborhood *R of v_0 is a simply-connected region in M whose boundary is a regular loop and whose only irregular vertex is v_0.*

Definition 5.3 *An* open separatrix *is an open and straight path whose interior vertices are regular and at least one of the end vertices is irregular. If both vertices are irregular, the separatrix is* double-sided*. Otherwise, it is* single-sided*. A* closed separatrix *is a loop that contains exactly one irregular vertex. In the remainder of the paper we are only interested in open separatrices. There are $l(v_0)$ separatrices emanating from v_0.*

The separatrices of v_0 divide any of its valid neighborhoods into $l(v_0)$ sectors, with each sector bounded by two separatrices and ∂R. A straight path inside a valid neighborhood R can intersect at most three sectors, resulting in three *pure segments*, i.e., a segment contained entirely inside one sector. A segment is *distance preserving* if the graph distance of any vertex on the segment to the irregular vertex v_0 is the same. Otherwise, the distance will monotonically increase or decrease along the segment, in which case the segment is referred to as *distance varying* (see Fig. 3). If there are at least two segments on a straight path, one of them must be distance preserving (the middle segment if there are three segments) while the others are distance varying. The length of the straight path is the number of edges on the path, and the distance of the path to the irregular singularity v_0 is the minimal graph distance of any vertex on the segment to v_0. Furthermore, any path intersecting the two separatrices bounding a sector must be distance preserving.

The following lemma is needed to prove Theorem 5.1:

Lemma 5.4 *Consider a convex region R that contains exactly one irregular vertex v_0 in its interior. Then $b_i = d_{i+1} + d_{i-1} - d_i$.*

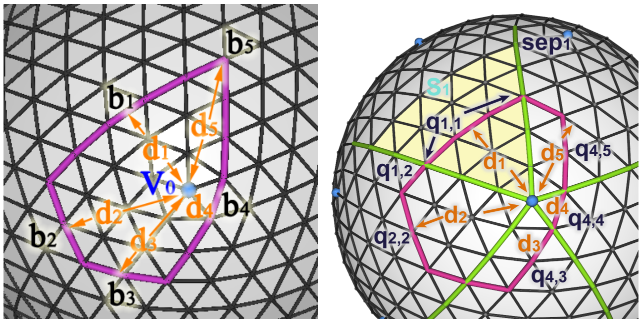


Figure 4: *Left: A convex pentagon enclosing a v5 vertex. The length of each side and its distance to the irregular vertex are denoted as b_i and d_i, respectively. Right: An illustration of variables used in the proof of Theorem 5.1. ∂R is shown as red loop and sector S_1 is shown in yellow. In this example $d_1 = d_5 = d_3 = 3$ and $d_2 = 4$; $q_{4,3}$ and $q_{4,5}$ have length 2. We also show how a side can intersect one, two, or three sectors.*

Proof From previous discussion we know that each sector can only intersect at most three sides of ∂R. Let $q_{i,j} = s_i \bigcap S_j$ where S_j is the j-th sector. If S_j only intersects one side, say s_i, then $q_{i,j}$ must be distance preserving and $length(q_{i,j}) = d_i$. If S_j intersects s_i and s_{i+1}, then exactly one of them is distance preserving and the other distance varying. We will assume s_i is distance preserving. Then $length(q_{i,j}) = d_{i+1}$ and $length(q_{i+1,j}) = d_i - d_{i+1}$. Finally, if S_j intersects sides s_{i-1}, s_i, and s_{i+1}, then $length(q_{i-1,j}) = d_i - d_{i-1}$, $length(q_{i+1,j}) = d_i - d_{i+1}$, and $length(q_{i,j}) = d_{i-1} + d_{i+1} - d_i$ (See Fig. 4 right).

Recall that side s_i can intersect at most three sectors. If s_i intersects exactly one sector, notice that the lemma is correct given the last equation from the last paragraph. If s_i intersects two sectors, then s_i will be distance preserving in exactly one (say sector S_j) and distance varying in the other (say sector S_{j+1}). Then $b_i = length(q_{i,j}) + length(q_{i,j+1}) = d_{i-1} + (d_{i+1} - d_i)$. If s_i intersects three sectors S_{j-1}, S_j, and S_{j+1}, then $q_{i,j}$ is distance preserving and $q_{i+1,j}$ and $q_{i-1,j}$ are distance varying. Consequently, $b_i = length(q_{i,j-1}) + length(q_{i,j}) + length(q_{i,j+1}) = (d_{i-1} - d_i) + d_i + (d_{i+1} - d_i) = d_{i-1} + d_{i+1} - d_i$. ∎

We now present the proof for Theorem 5.1.

Proof Given Lemma 5.4, we know that any convex region R satisfying the condition of this theorem will have $l(v_0)$ sides and the length and distance of these sides satisfy

$$b_i = d_{i-1} + d_{i+1} - d_i \tag{3}$$

for all $0 \le i < l(v_0)$. Assume that we have retriangulated the interior of region R such that there is a unique irregular vertex v_0'. Let d_i' be the distance of v_0' to side s_i. Then we have

$$b_i = d_{i-1}' + d_{i+1}' - d_i' \tag{4}$$

In other words, the configurations of the interior of R before and after the retriangulation satisfy the same system of equations $Ad = b$ where $d = [d_1, ..., d_{l(v_0)}]$, $b = [b_1, ..., b_{l(v_0)}]$, and $A = a_{ij}$ in which $a_{ij} = 0$ except $a_{ii} = -1$ and $a_{i,i+1} = a_{i+1,i} = 1$. Notice that A has a non-zero determinant if $l(v_0)$ mod $6 \neq 0$, which means that there is a unique solution for d_i. Consequently, the retriangulations before and after represent the same configuration and the irregular vertex cannot be moved. If the valence is a multiple of six the determinant of A is 0 and multiple solutions exist. ∎

While Theorem 5.1 only states that moving a single irregular vertex within a convex region is impossible, we have not found a case where we can move an irregular vertex in practice. In fact, the requirement in the theorem that the enclosing polygon is convex can be relaxed. Basically, if the polygon can be enlarged to a convex one without including any additional irregular vertex, then it is impossible to move the irregular vertex in the original polygon which may be concave.

Similarly, we have found that it is in general impossible to cancel a pair of irregular vertices with opposite indexes as stated in the following theorem.

Theorem 5.5 *Consider a convex region R that contains exactly two irregular vertices v_1 and v_2 in its interior. If $l(v_1)+l(v_2)=12$, i.e., v_1 and v_2 have opposite discrete Gauss curvatures, it is impossible to retriangulate the interior of R to have a different configuration that is free of irregular vertices.*

Without loss of generality, we will assume that $l(v_1) < l(v_2)$. Furthermore, we are not interested in the degenerate cases with $l(v_1) \leq 2$. The proof of this theorem depends on the following lemmas.

Lemma 5.6 *Consider a convex region R that contains no irregular vertex v_0 in its interior. The lengths of the six sides of ∂R satisfy that $b_1+b_2=b_4+b_5$, $b_2+b_3=b_5+b_6$, and $b_3+b_4=b_6+b_1$.*

Proof We select an arbitrary interior vertex v_0 which must have a valence of 6. It is straightforward to verify $b_i = d_{i-1}+d_{i+1}-d_i$ using argument similar to that for Lemma 5.4. Consequently, $b_1+b_2=(d_6+d_2-d_1)+(d_1+d_3-d_2)=(d_3+d_5-d_4)+(d_4+d_6-d_5)=b_4+b_5$. Similarly we have $b_2+b_3=b_5+b_6$ and $b_3+b_4=b_6+b_1$. ∎

For convenience, we will express the three equations from Lemma 5.6 simply as $Ab_R=0$ where $A=\begin{pmatrix} 1 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 & -1 \\ -1 & 0 & 1 & 1 & 0 & -1 \end{pmatrix}$ and $b_R=(b_1,b_2,...,b_6)$.

We now prove Theorem 5.5 by showing that $Ab_R \neq 0$ if the convex region R contains an irregular vertex pair. In the main part of the paper, we will only show that this property is true for the smallest enclosing convex hexagon K for the irregular vertex pair. This part is interesting because it illustrates the different possible configurations. The remaining two parts are that the smallest enclosing hexagon is unique and that any enclosing convex hexagon R for the irregular vertex pair can be reduced to K and $Ab_R = Ab_K$. These two parts are shown in the appendix.

Lemma 5.7 *Under the assumption of Theorem 5.5, the smallest enclosing convex hexagon K for v_1 and v_2 satisfies that $Ab_K \neq 0$.*

Proof Figure 5 illustrates the minimal enclosing polygons for an irregular vertex pair. If the irregular vertex pair is connected by a separatrix, it is straightforward to verify that the minimal enclosing polygon K is the 1-ring neighborhood of the separatrix (Figure 5 (left column)).

If v_1 and v_2 are not connected by a separatrix, note that v_1 is in a sector of v_2 and v_2 is in a sector of v_1 since they are not connected by a separatrix. The intersection of the two sectors is a parallelogram with sides of lengths L_1 and L_2, respectively.

If $l(v_1)=5$ and $l(v_2)=7$ (Figure 5 (2a)) and if $l(v_1)=4$ and $l(v_2)=8$ (Figure 5 (2b)), we can show that ∂K is the 1-ring neighborhood of the parallelogram.

If $l(v_1)=3$ and $l(v_2)=9$ (Figure 5 (2c)), ∂K includes four sides of ∂N_2, and the extensions of two more sides of ∂N_2 which intersect after winding around N_1 from opposing sides. We denote the 1-ring neighborhood of v_1 and v_2 as N_1 and N_2, respectively. The fact that the extensions of two sides of ∂N_2 can intersect is a result of the triangulation around v_1. We omit the details in the interest of space.

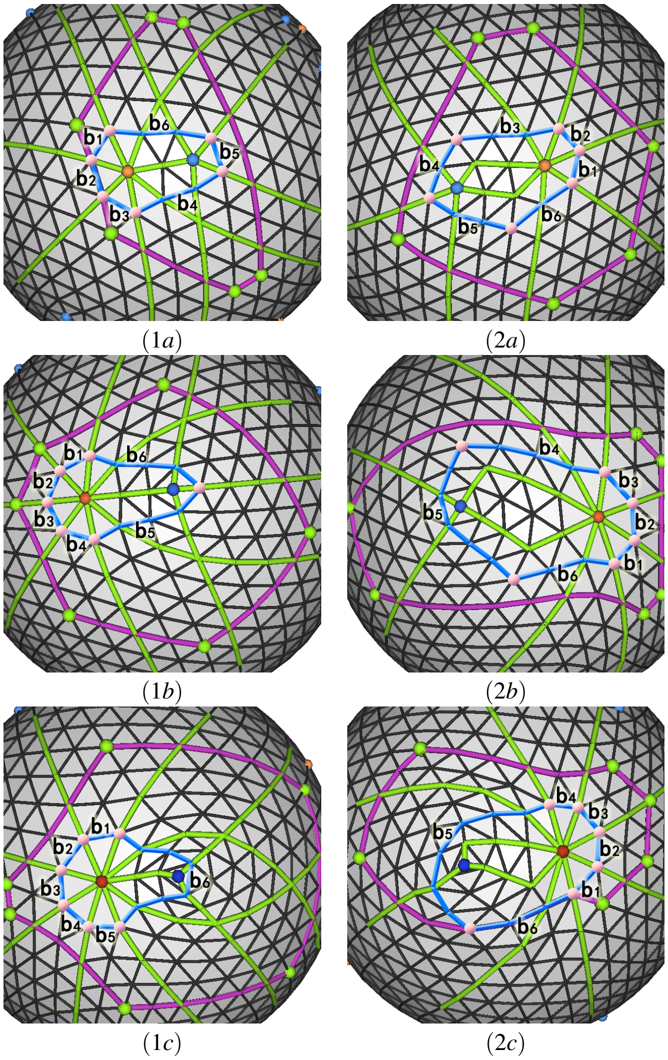


Figure 5: *Minimal enclosing convex hexagon for an irregular vertex pair: (a) v5/v7, (b) v4/v8, and (c) v3/v9. In the left column, the pair is connected by a separatrix, while the pair in the right column is not. The minimal enclosing polygon is colored in blue.*

From the above characterization of K (also see Figure 5) it is clear that $b_i=1$ for at least two i's which are sides of N_2. We will find the first such side in a counterclockwise fashion and name it s_1. We will then label other sides incrementally. Consequently, $b_{K,1}+b_{K,2}=2$. Notice that with the exception of v3/v9 pair connected by a separatrix, in all other cases either $b_{K,4}$ or $b_{K,5}$ or both are larger than 1 due to the separation between the irregular vertex pair. Consequently, $b_{K,4}+b_{K,5} > 2 = b_{K,1}+b_{K,2}$. Thus $Ab_K \neq 0$ in these five cases. In the case of v3/v9 connected by a separatrix, we have $b_{K,i}=1$ for $1 \leq i \leq 5$ and $b_{K,6}>1$. Consequently, $b_{K,2}+b_{K,3}=2<b_{K,5}+b_{K,6}$ and therefore $Ab_K \neq 0$. ∎

Lemma 5.8 *Under the assumption of Theorem 5.5, the smallest enclosing convex hexagon for v_1 and v_2 is unique (see appendix for the proof).*

Lemma 5.9 *Under the assumption of Theorem 5.5, any enclosing convex polygon R for v_1 and v_2 satisfies that $Ab_R = Ab_K \neq 0$ (see appendix for the proof).*

We now give the proof of Theorem 5.5.

Proof Given a convex polygon R enclosing an irregular vertex pair

v_1 and v_2 where $l(v_1)+l(v_2)=12$, we have $Ab_R \neq 0$ based on Lemma 5.9. Assume that it is possible to retriangulate the interior of R such that it no longer contains any irregular vertex. Call the new triangulated region R'. According to Lemma 5.6 we have $Ab_{R'}=0$. However, $b_R=b_{R'}$ as the retriangulation does not change the boundary connectivity. Consequently, we have $0=Ab_{R'}=Ab_R \neq 0$, a contradiction. ▮

Additionally, an irregular vertex pair cannot be canceled inside any sub-region of an enclosing convex hexagon. Consequently, an irregular vertex pair cannot be canceled inside any region that can be enlarged to a convex hexagon by adding only regular vertices. Furthermore, b_i of any enclosing convex hexagon satisfy formulas similar to but not the same as those from the irregular vertex movement case. We did not count on such facts for the proof of Theorem 5.5 but believe they are valuable.

Together, Theorems 5.1 and 5.5 demonstrate that irregular vertex editing is a much different and possibly more challenging problem than topological control in vector and tensor field processing. We need to identify new fundamental operations for the control of the type, location, and number of irregular vertices.

6 Fundamental Semantic Editing Operations

In this section we lay the groundwork for the (more powerful) composite semantic editing operations. First, we review the graph editing operations (edge flip, edge collapse, and vertex split) that are the most basic atomic primitives that we employ to edit a mesh. These operations are local, but they do not have an intuitive semantic interpretation when it comes to mesh connectivity editing. Therefore, the second subsection introduces fundamental semantic operations for v5 and v7 vertices. These fundamental semantic operations are achieved by one or multiple graph editing operations. They have local influence and enable a semantic interpretation to the graph editing operations. Similarly, we introduce two classes of operations to reduce any irregular vertex to a set of v5 or v7 vertices in the third subsection.

6.1 Graph Editing Operations

We selected edge flip, edge collapse, and vertex split as graph editing operations because they are relatively simple and robust and allow constructing all fundamental and composite semantic editing operations with systematic combinations of these operations. However, we have found that the influence of these operations on vertex valence is not very intuitive and difficult to understand. We therefore provide a semantic interpretation for these edits in the next subsection.

Edge Flip: There are four vertices involved in an edge flip. After an edge flip, the valence of two vertices will be increased by 1, and the valence of the other two vertices will be decreased by 1. The inverse operation of an edge flip is another edge flip of the same edge.

Edge Collapse: There are four vertices involved in an edge collapse. After an edge collapse, the valence of two vertices will be decreased by 1, the valence of one vertex w is $d(w)=d(u)+d(v)-4$ ((u,v) is the collapsed edge, w is the remaining vertex), and one vertex will be deleted.

Vertex Split: There are four vertices involved in a vertex split. To define a vertex split, two edges incident to the split vertex w need to be selected. The two incident edges separate the remaining edges incident to w in two groups containing d_1 and d_2 edges. After a vertex split, the valence of two vertices will be increased by 1, the valences of the other two vertices u, v are $d(u)=d_1+3$, $d(v)=d_2+3$.

6.2 Atomic Semantic Editing Operations

We now describe five atomic fundamental semantic editing operations which can be combined to achieve the higher-level composite semantic operations for v5 and v7 vertices. The challenge in this section is to give all operations the correct semantic interpretation. While the final results are simple, the proposed structure and semantics are very important for understanding the proposed operations.

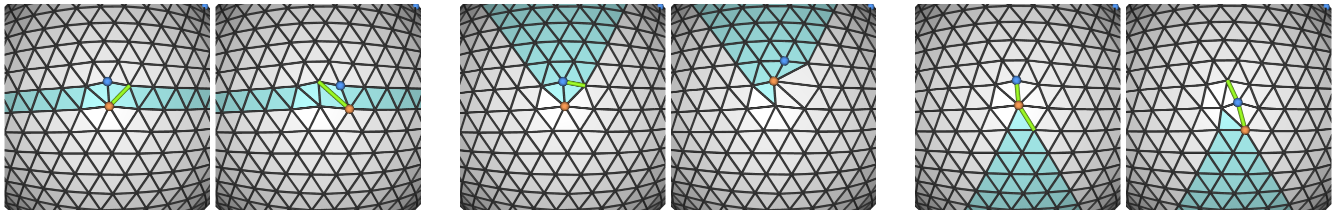
Atomic $5-7$ Movement: A $5-7$ irregular vertex pair that is connected by an edge is moved. This operation corresponds to applying the three graph editing operations (edge flip, edge collapse, and vertex split) to an existing $5-7$ pair. Each graph editing operation can be applied in two different configurations resulting in six possible movement directions. Assuming surrounding vertices are regular the directions correspond to the six edges per regular vertex. We show one configuration for each graph editing operation in Fig. 6. For example, in Fig. 6 (a) we show an irregular vertex pair plus the selected edge before and after the flipping operation. Performing an edge flip will make the original irregular vertex pair into regular vertices and two neighboring vertices into a new $5-7$ pair. Notice that the newly created irregular vertices are still adjacent to each other, and the edge connecting them is parallel to the edge connecting the original irregular vertex pair before the move. Continuing edge flip operations will have the effect of parallel transporting the $5-7$ pair in the zone shown in cyan. This is similar to the notion of *parallel transport* of a vector (the edge) along the mesh surface. Performing appropriate edge collapses and vertex splits will have similar effects as shown in Fig. 6 (b, c). Moreover, existing triangles can be eliminated during edge collapses and new triangles can be created during vertex splits. With a combination of the three basic operations the $5-7$ pair can move anywhere on the mesh.

Atomic v5 (v7) Movement and $5-7$ Generation: We briefly explain this operation for a v5 vertex because the operation on a v7 vertex is similar. A v5 vertex can move one step in one of six directions. As a side effect an adjacent $5-7$ pair is generated. This operation corresponds to applying one of the three graph editing operations (edge flip, edge collapse, and vertex split) to an existing v5 vertex (see Fig. 7). In Fig. 7 (a) a single v5 vertex is moved by an edge flip. Since there are two new v5 vertices we have to pick one as the moved v5 vertex and the other one as part of the generated $5-7$ pair. This means that exactly the same edge flip can have two different semantic interpretations based on which v5 vertex is picked. Knowing the correct semantic interpretation of this proposed atomic operation is a key insight necessary to build higher-level semantic operations.

Atomic v5 (v7) Movement and $5-7$ Removal: This fundamental operation is the inverse of the operation described above. It can be understood by reading Fig. 7 in the reverse direction and replacing edge collapses by vertex splits.

Atomic Two v5 and Two v7 Generation: A regular vertex is split into a $5-5-7-7$ irregular vertex quadruple that is contained in two triangles. This operation corresponds to applying the three graph editing operations (edge flip, edge collapse, and vertex split) to a regular triangulation (see Fig. 8). Interestingly, one configuration is generated by two subsequent graph editing operations.

Atomic Two v5 and Two v7 Removal: This fundamental operation is the inverse of the operation described above. It can be understood by reading Fig. 8 in the reverse direction.

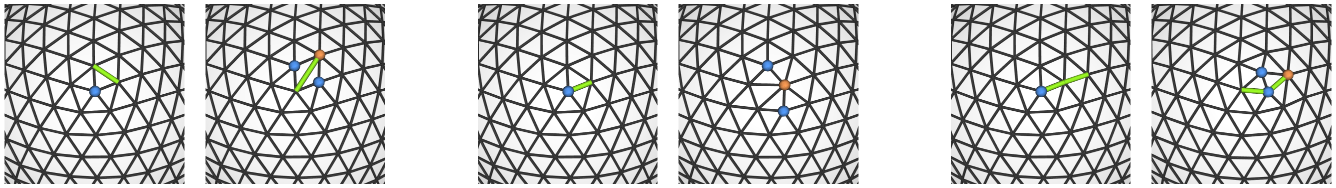


(a) Translating a $5-7$ pair in the mesh by edge flips. The cyan region indicates where the pair can move. The number of triangles is not changed.

(b) Translating a $5-7$ pair in the mesh by edge collapse. The cyan region indicates where the pair can move. A triangle strip is deleted through subsequent edge collapse operations.

(c) Translating a $5-7$ pair in the mesh by vertex split. The cyan region indicates where the pair can move to. A new triangle strip is created.

Figure 6: *Atomic $5-7$ irregular vertex movement. v5 vertices are shown in blue and v7 vertices in orange.*



(a) Edge Flip

(b) Edge Collapse

(c) Vertex Split

Figure 7: *Atomic v5 movement and $5-7$ pair generation: v5 vertices are shown in blue and v7 vertices in orange.*

6.3 Fundamental Problem Reduction Operations

There are two classes of operations to reduce the editing problem to v5 and v7 vertices. We call them vertex type splitting and vertex type aggregation.

Type Splitting: There are two classes of operations. The first class of operations splits a vk $(k > 7)$ vertex into $k-6$ v7 vertices. The second class of operations splits a v4 (or v3) vertex into two (or three) v5 vertices. There are many possible configurations and we show three examples in Fig. 9.

Type Aggregation: There are two classes of operations. The first class combines multiple v7 vertices into one vk $(k > 7)$ vertex. The second class of operations combines two (or three) v5 vertices into one v4 (or v3) vertex. This operation is the inverse of type splitting.

7 Composite Semantic Editing Operations

We describe the algorithms and constraints for the composite semantic editing operations. We consider movement, removal, and generation in the following three subsections.

7.1 Irregular Vertex Movement

In this section we describe operations for moving a pair of irregular vertices, i.e., an $m-n$ pair. There are three categories: 1) $m,n > 6$, 2) $m,n < 6$, and 3) $m < 6 < n$. Through splitting and merging of irregular vertices, we can reduce these to three fundamental cases: $7-7$, $5-5$, and $5-7$. We will provide the detail for the last case, i.e., $5-7$, since the other two are fundamentally similar.

7.1.1 Complex Topology Editing: Moving $5-7$ Pairs

Here we describe the general case of a moving $5-7$ pair when the two vertices are not adjacent. We can first construct a simple path between the two irregular vertices by finding the intersection of two separatrices, each from one irregular vertex. The intersection point will be the turn point unless the separatrices completely overlap. To move the pair, we make use of the following three-step pipeline:

1. Apply the atomic v5 one step movement and $5-7$ pair generation operation to move the v5 vertex in the desired direction.
2. Move the adjacent $5-7$ pair towards the original v7 vertex through a sequence of atomic $5-7$ movement operations, until the pair and the v7 vertex are adjacent.
3. Apply the atomic v7 one step movement and $5-7$ pair removal operation to convert the triple so that only one v5 vertex is left, at a shifted location.

The final result is that the old $5-7$ pair (u,v) is moved to a new location (u',v'). The relative position of the two vertices remains the same. If we use the different combinations of the basic operations, we can move the pair in different directions. Since some of the moving directions are redundant, there are 6 different directions in total (Fig. 10). The movement is explained in the caption. Alternatively, the reverse sequence can be started from the v7 vertex.

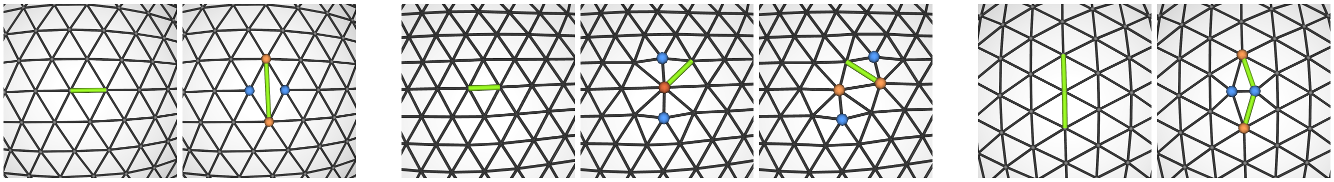
7.1.2 Complex Topology Editing: Moving $5-5$ and $7-7$ Pairs

Moving a pair of $5-5$ or $7-7$ irregular vertices (not necessarily adjacent) is similar to moving a $5-7$ pair. As with a simply connected or complex connected $5-7$ pair, there are 6 different directions of movement. However, in contrast to the purely translative movement of the $5-7$ pair, after the movement, the relative location of the irregular vertices will be changed.

We explain the possible movements by the example shown in Fig. 11 left. There are two irregular vertices of valence 5, which are connected by 2 simple paths. Each of the paths consists of two segments of length L_1 and L_2. We can select an arbitrary quad loop, with side lengths $b_1,\ldots,b_4$, and the distance between the sides and irregular vertices can be specified by $d_1,\ldots,d_4$. We have the following set of equations: $b_i = d_{i-1} + d_{i+1} - d_i + L_p$ for $1 \le i \le 4$ where $p = 1$ if $i = 1,3$ and $p = 2$ if $i = 2,4$.

The solution of these equations is not unique. The general solution is: $d_1 = t_1$, $d_2 = t_2$, $d_3 = b_1 - b_3 + t_1$, $d_4 = b_2 - b_4 + t_2$, $L_1 = b_1 + b_4 - b_2 + t_1 - 2t_2$, and $L_2 = b_2 + b_3 - b_1 + t_2 - 2t_1$.

From this solution, we know that if we change d_1 and d_2 (move one irregular vertex), d_2 and d_4 will change correspondingly (the other irregular vertex will also be moved). Additionally, since $\Delta d_1 = \Delta d_3$ and $\Delta d_2 = \Delta d_4$, the movement of the two irregular points will be symmetric. In other words, the turn point on the path connecting the two irregular vertices may slide along the path, leaving the im-

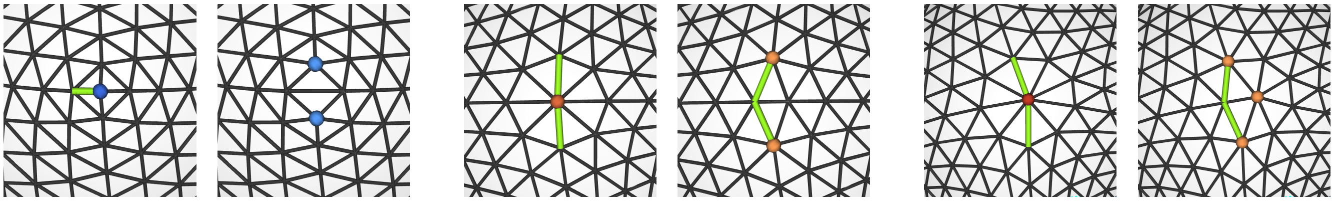


(a) Edge Flip

(b) Edge Collapse and Edge Flip

(c) Vertex Split

Figure 8: *Atomic two v5 and two v7 generation: v5 vertices are shown in blue, v7 vertices in orange, and the v8 vertex in red.*



(a) Splitting a v4 vertex

(b) Splitting a v8 vertex

(c) Splitting a v9 vertex

Figure 9: *Examples of type splitting operations. Higher order irregular vertices are reduced to multiple v5 or v7 vertices.*

pression the two end vertices (irregular) are approaching or leaving each other as well as rotating about the middle point between the two vertices (not necessarily on the path). We refer the readers to the accompanying video for a live demonstration.

7.2 Irregular Vertex Removal

The most common irregular vertex removal operations are $5-5-7$ or $5-7-7$ triple cancellation. Without losing generality, we discuss the case of $5-5-7$, as the $5-7-7$ triple cancellation is almost identical. In this section we describe how to cancel triples that are not adjacent to each other. We select a $5-7$ pair (say u and v) and move it towards w (5 vertex) until v (valence 7) is adjacent to w (valence 5). We then move the $5-7$ pair of v and w towards u until either v or w or both are adjacent to u. We then apply the atomic v5 movement and $5-7$ removal operation to convert the triple into a 5 vertex and thus complete the operation. The second high-level operation is to cancel a $5-5-7-7$ quadruple. This operation is described in the next section as it is more important for generation than for removal.

7.3 Irregular Vertex Generation

We consider two operations. The first is the inverse of $5-5-7$ or $5-7-7$ triple cancellation described in the previous subsection. The second operation is to split a regular vertex into a $5-5-7-7$ irregular vertex quadruple. We always generate at least two $5-7$ pairs. But these two pairs are not arbitrary since they must satisfy some constraints. See Fig. 11 for an illustration of this configuration. We denote the four vertices by v_1, v_2, v_3, v_4, and connect v_i to v_{i+1} by the shortest complex path p_i that walks clockwise. Such a path p_i is composed by two straight paths, whose lengths are x_i and y_i. The pair (x_i, y_i) can denote the distance between v_i and v_{i+1}. All the paths form a convex polygon, whose opposite sides must have the same length. In other words, $x_i = x_{i+2}$, $y_i = y_{i+2}$, for $i = 1, \ldots, 4$. The proof is based on employing the argument of contradiction, which we do not include in the paper in the interest of space. Otherwise, we could remove one $5-7$ pair and we would have generated a single $5-7$ pair from a regular triangulation. This is impossible according to our proof in Section 5.

8 Geometric Operations

Here we discuss relaxation operations on the mesh geometry.

Global Relaxation: The connectivity editing operations often result in triangles with poor aspect ratios. Therefore, we use a global relaxation to make the length of edges in the mesh uniform and the angles around each vertex divided into equal partitions. To achieve this, we can move the vertices to new locations on the surfaces such that an energy term is minimized. Let p_i be the position of vertex v_i, N_i be the set of neighbors of vertex v_i, d be the average length of all edges, and u_{ij} be the unit length target tangent vector. The energy term is defined as follows:

$$E(p) = \sum_i \sum_{j \in N_i} \| p_j - (p_i + du_{ij}) \|^2. \tag{5}$$

Suppose the tangent plane at vertex position p_i is P. If we project the vector $\overrightarrow{p_ip_j}$ to P, we get another vector $\overrightarrow{p_ip_j}_p$. The angle between $\overrightarrow{p_ip_j}_p$ and the local x direction P_x is denoted by α_{ij}^0. Our goal is that the tangent space is equally partitioned angularly by these tangent vectors so that we can calculate the target angle α_{ij} between the target tangent vector u_{ij} and P_x. Let k_j denote the index of the v_j sorted in counterclockwise order, and α_0 denote the angle between the target tangent vector with index 0 and vector P_x. Then the angle between α_{ij} and α_0 is $\beta_{ij} = k_j(2\pi/l(v_i))$. We can calculate the best $\alpha_0 = \sum_{p_j \in N_i} (\alpha_{ij}^0 - \beta_{ij})/l(v_i)$, so $\alpha_{ij} = \alpha_0 + \beta_{ij}$. After we calculate α_{ij}, u_{ij} can be calculated by rotating P_x around the normal at v_i by an angle of α_{ij}.

The constraint that all vertices are on the surface of mesh M can be approximated by the constraint that all vertices lie on their previous plane. After the location of a vertex is changed, it is projected to the surface of the mesh again. This constraint is linear in every step, which leads to a quadratic energy function that can be solved by quadratic solvers, such as *UMFPACK*.

Local Relaxation: Local relaxation is a simpler smoothing algorithm where vertices are moved one at a time [Turk 1992]. We implemented global and local relaxation in our framework. For our results we ended up using local relaxation more often simply because it is faster.

9 Interface

User interface: We implemented all modeling operations described in the paper. We provide tools for selecting sets of vertices, faces, and edges. In our experience most editing time is spent using the vertex pair movement operations. Our interface allows the user

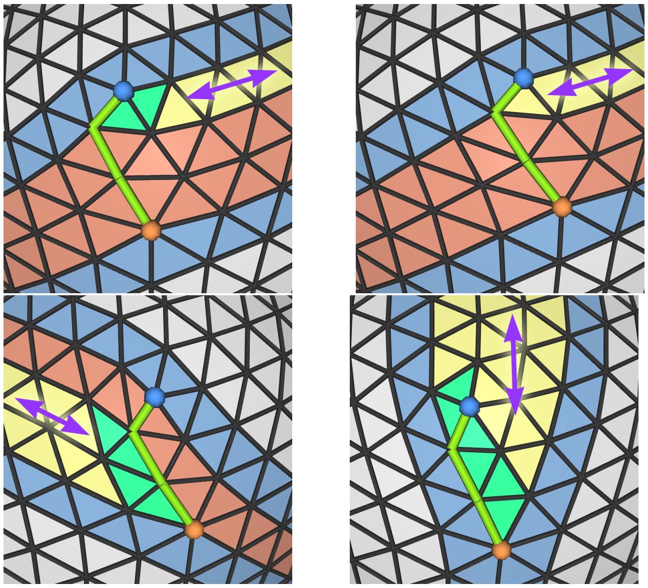


Figure 10: *Moving a $5-7$ vertex pair. In the top left and the top right we show how the vertex pair moves one step to the right (configuration one). In the bottom left and bottom right we show configurations two and three. The blue triangle strips remain unchanged. The purple arrows show the two possible movement directions for each configuration. The yellow strip is either removed or extended depending on which direction the $5-7$ pair moves. The red strips are modified by edge flips. The green triangles will be generated or removed depending on the direction of the movement. In the general case, the number of triangle strips affected is determined by the length of the connecting path and the position of the turning point.*

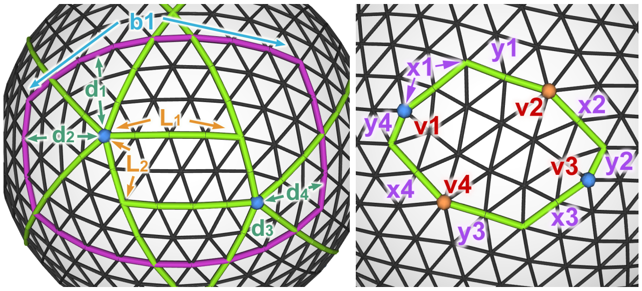


Figure 11: *Left: moving a $5-5$ vertex pair. Right: irregular vertex generation.*

to select a vertex pair and one of six movement directions of one selected vertex. The movement of the other vertex is determined. According to our analysis of possible and impossible editing operations this is the most intuitive composite editing operation that gives the user full control over the outcome. Movement operations are interleaved with local and global relaxation. Selected editing operations are demonstrated in the video accompanying the paper.

Example Application: We visualize the impact of irregular vertices using pattern design as example application. We implemented manual and procedural tools for assigning material IDs to triangles and edges and to replace triangles with more complex geometry. This application is inspired by recent papers in architectural geometry, e.g. [Liu et al. 2006; Schiftner et al. 2009]. The goal is to demonstrate that it is not sufficient to optimize vertex locations. If the mesh edges are part of the design, the irregular vertices will have a strong visual impact.

10 Application & Result

In this section we will describe modeling results for four selected models. The input models were preprocessed by a greedy retriangulation algorithm and also smoothed with the algorithm described in the paper. As a result all input models have a reasonably nice geometry. If a general mesh would be taken as input, the geometry would become locally smooth at the locations of the edits. Therefore, our current main limitation is that we cannot retain sharp features during editing. It would be necessary to implement additional geometry processing operations, e.g. feature-preserving relaxation algorithms. The results consist of two parts: the editing of irregular vertices and the application of a pattern. The patterns were chosen to motivate different goals for editing the irregular vertices.

Duck: We designed a star pattern on a duck with approximately 10K triangles. The input mesh has 160 irregular vertices (86 v5 and 74 v7) and the output mesh has 46 irregular vertices (29 v5 and 17 v7). The input mesh, the designed mesh, and the applied pattern are shown in Fig. 12. The goals of the design were as follows: 1) reduce the number of irregular vertices to smooth the mesh, 2) put irregular vertices close to regions of high curvature, and 3) ensure that several v5 vertices do not have other irregular vertices nearby so that a star pattern can be seeded from these vertices.

Bunny: We designed a zebra pattern on a bunny with approximately 40K triangles. The input mesh has 1265 irregular vertices (1 v4, 639 v5, 621 v7, and 4 v8) and the output mesh has 84 irregular vertices (48 v5 and 36 v7). The input mesh, the designed mesh, and the applied pattern are shown in Fig. 13. The zebra pattern requires that singularities are placed according to aesthetic criteria.

Souzou Model: We designed a circle pattern on a U-shaped model with approximately 2500 triangles (Fig. 14). The interesting characteristic of this pattern is that many irregular vertices are generated intentionally to create circles with the triangulation. The input mesh has 36 irregular vertices (2 v4, 21 v5, and 13 v7). The first editing step was to reduce the irregular vertices to a minimum of 12 v5 vertices. Subsequently many irregular vertices are added to form the circle pattern. The final mesh has 3 v4, 282 v5, and 276 v7 irregular vertices.

Architectural Model: We designed a stripe patterns on the 2-peak architectural model with approximately 10K triangles (Fig. 1). The input mesh has 278 irregular vertices (10 v4, 133 v5 vertices, 132 v7, and 3 v8) and the output mesh has 24 irregular vertices (18 v5 vertices and 6 v7). We also show what happens if the stripe pattern is applied to the original model. First, in the edited model we can ensure that stripes have only a width of one, but in the original model we have stripes of width one and two. Second, in the edited model the shapes between stripes are simple hexagons and pentagons, but in the original model the shapes can be fairly complex. Finally, the stripes are smoother in the edited model, but in the original model they make sharper turns.

11 Conclusion

In this paper we introduce the problem of explicit mesh connectivity optimization for the control of irregular vertices. We also present to our knowledge the first system that provides control over the type, location, and number of irregular vertices in a triangle mesh. As part of this research, we present a supporting theoretical analysis as well as identify a comprehensive set of operations that provides sufficient flexibility to the user. We have also developed a taxonomy of operations, at different levels. We provide efficient composite semantic editing from a set of fundamental operations, which themselves are implemented with graph-level editing opera-

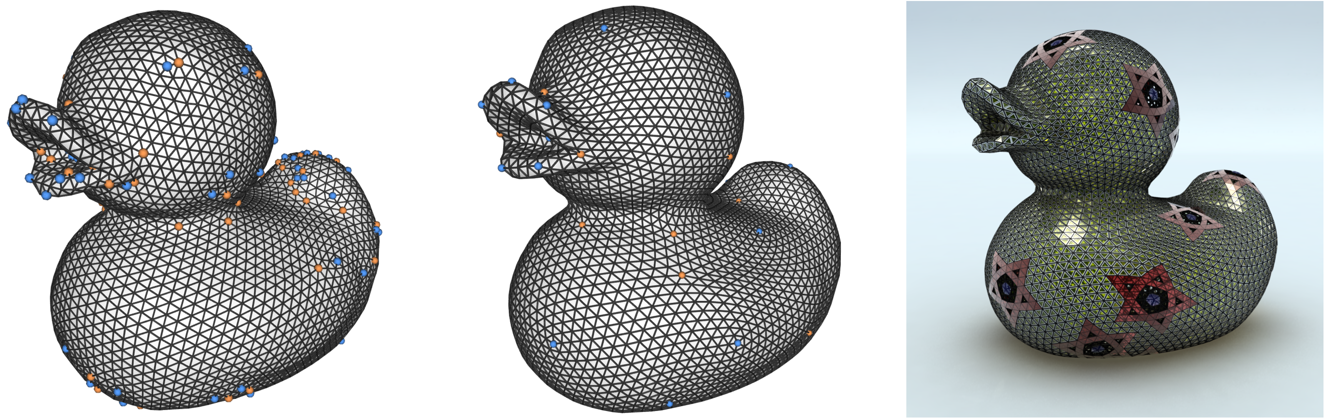


Figure 12: *From left to right: input model, edited model, star pattern.*

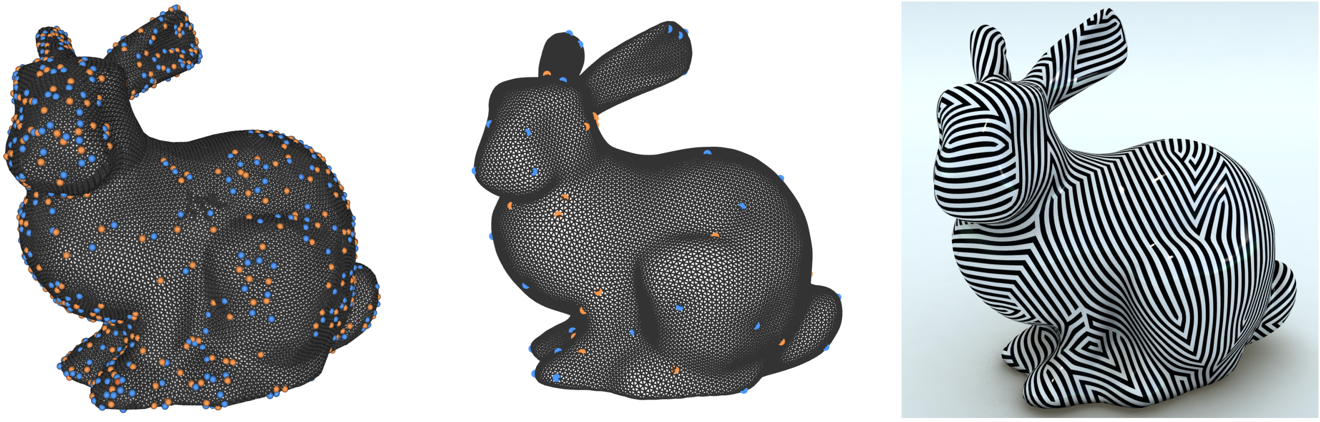


Figure 13: *From left to right: input model, edited model, zebra pattern.*

tions. We demonstrate the usefulness of our approach with a number of examples in pattern design and architectural modeling.

In the future we wish to explore adapting the system to the editing of quad meshes. We also plan to investigate how to adapt our system for remeshing purposes, with the possibility of performing geometry-aware mesh editing.

Acknowledgements

We would like to acknowledge the help of the reviewers for improving the proofs and Bao Fan and Christopher Grasso for help with the figures. This work was funded by NSF contracts IIS-0757623, IIS-0915990, CCF-0643822, CCF-0830808, and IIS-0917308.

References

Akleman, E., and Chen, J. 2006. Practical polygonal mesh modeling with discrete gaussian-bonnet theorem. In *GMP*, 287–298.

Alliez, P., Meyer, M., and Desbrun, M. 2002. Interactive geometry remeshing. *ACM Transactions on Graphics 21*, 3, 347–354.

Alliez, P., de Verdière, É. C., Devillers, O., and Isenburg, M. 2003. Isotropic surface remeshing. In *Proceedings of Shape Modeling International*, 49–58.

Bommes, D., Zimmer, H., and Kobbelt, L. 2009. Mixed-integer quadrangulation. *ACM Transactions on Graphics (TOG) 28*, 3, 77.

Desbrun, M., Meyer, M., Schröder, P., and Barr, A. H. 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH 1999*, 317–324.

Fisher, M., Schröder, P., Desbrun, M., and Hoppe, H. 2007. Design of tangent vector fields. *ACM Transactions on Graphics (TOG) 26*, 56.

Gu, X., Gortler, S. J., and Hoppe, H. 2002. Geometry images. *ACM Trans. Graph. 21*, 3, 355–361.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W. 1993. Mesh optimization. In *SIGGRAPH 1993*, ACM, New York, NY, USA, 19–26.

Hoppe, H. 1996. Progressive meshes. In *SIGGRAPH 1996*, ACM, New York, NY, USA, 99–108.

Lewiner, T., Lopes, H., Medeiros, E., Tavares, G., and Velho, L. 2010. Topological mesh operators. *Computer Aided Geometric Design 27*, 1, 1–22.

Liu, Y., Pottmann, H., Wallner, J., Yang, Y.-L., and Wang, W. 2006. Geometric modeling with conical meshes and developable surfaces. *ACM Transactions on Graphics 25*, 3, 681–689.

Nealen, A., Sorkine, O., Alexa, M., and Cohen-Or, D. 2005. A sketch-based interface for detail-preserving mesh editing. *ACM Trans. Graph. 24*, 3, 1142–1147.

Palacios, J., and Zhang, E. 2007. Rotational symmetry field design on surfaces. *ACM Transactions on Graphics 26*, 3, 55.

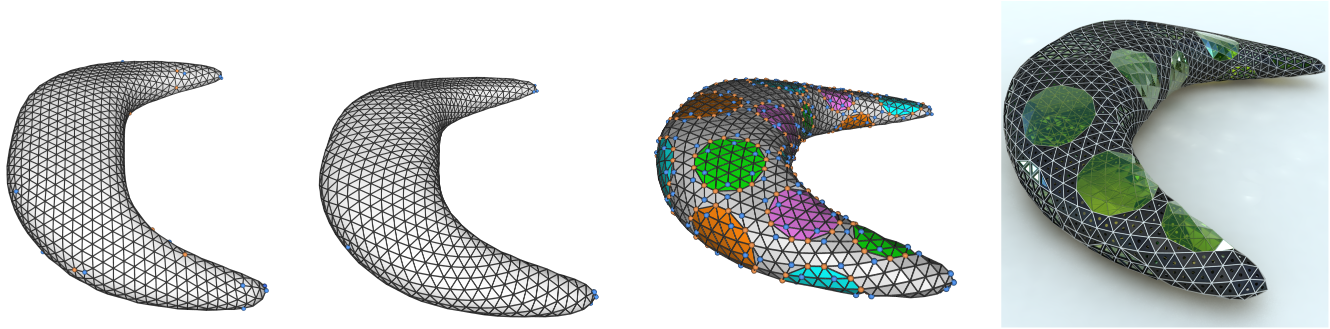


Figure 14: *From left to right: a) input model, b) edited model to obtain the minimum number of* 12 *v5 vertices, c) edited model with many irregular vertices to form circles on the surface, and d) the rendered model with replaced geometry.*

RAY, N., VALLET, B., LI, W. C., AND LÉVY, B. 2008. N-symmetry direction field design. *ACM Trans. Graph. 27*, 2, 1–13.

SCHIFTNER, A., HÖBINGER, M., WALLNER, J., AND POTTMANN, H. 2009. Packing circles and spheres on surfaces. *ACM Trans. Graph. 28*, 5, 1–8.

SURAZHSKY, V., AND GOTSMAN, C. 2003. Explicit surface remeshing. In *2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 30.

SURAZHSKY, V., ALLIEZ, P., AND GOTSMAN, C. 2003. Isotropic remeshing of surfaces: a local parameterization approach. In *Proceedings of 12th International Meshing Roundtable*.

TAUBIN, G. 2002. Detecting and reconstructing subdivision connectivity. *The Visual Computer 18*, 5–6.

TURK, G. 1992. Re-tiling polygonal surfaces. *SIGGRAPH 1992*, 55–64.

ZHANG, E., MISCHAIKOW, K., AND TURK, G. 2006. Vector field design on surfaces. *ACM Transactions on Graphics 25*, 4, 1294–1326.

ZHANG, E., HAYS, J., AND TURK, G. 2007. Interactive tensor field design and visualization on surfaces. *IEEE Transactions on Visualization and Computer Graphics 13*, 1, 94–107.

Appendix

Lemma 5.8: Under the assumption of Theorem 5.5, the smallest enclosing convex hexagon for v_1 and v_2 is unique.

Proof We first make the following comment. A regular polygon R (no irregular vertices on ∂R) is convex if and only if for any vertices $u, v \in R$ any shortest path connecting them is also in R. A sketch of the proof is as follows. Assume that the polygon is convex yet there exist $u, v \in R$ such that a shortest path γ connecting u and v is partially outside R. Then γ must intersect ∂R an even number of times. Consider the first two such intersection points w_1 and w_2. Then there must be a negative angle of turn on ∂R between w_1 and w_2. This contradicts that R is convex. For the other direction of the comment, assume that R is not convex yet for any $u, v \in R$ we have $\gamma \subset R$ where γ is any shortest path connecting u and v. In this case we can find two sides of ∂R that have a negative turn since R is concave. Let the two sides be s_i and s_{i+1} and the point of turn is w. Consider the vertices on ∂R immediately before and after w. They are located on s_i and s_{i+1}, respectively. Between these vertices there is a shortest path connecting them but is outside R, again a contradiction.

Given the comment, it is straightforward to show that the minimal enclosing convex polygon is unique. Assume there are two such minimal polygons $U \neq V$. Then $W = U \bigcap V$ must be strictly smaller than U or V. Say W is strictly smaller than U. Furthermore, since U and V are convex they both satisfy the condition that for any vertices $u, v \in R$ any shortest path connecting them is also in R. Consequently, W satisfies this condition as well yet it is smaller than U. Thus U is not minimal which is a contradiction. ∎

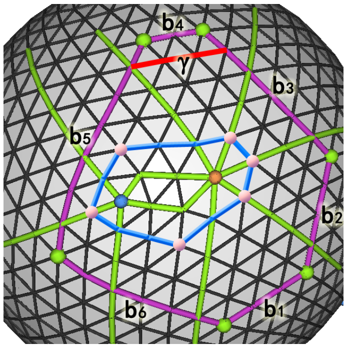


Figure 15: *An example shrinking operation.*

Lemma 5.9: Under the assumption of Theorem 5.5, any enclosing convex polygon R for v_1 and v_2 satisfies that $Ab_R = Ab_K \neq 0$.

Proof Let $d_R = \sum_{i=1}^{6} d_{R,i}$ where $d_{R,i} = dist(s_{R,i}, K)$ is the distance between side i of ∂R and K. We will use mathematical induction on d_R.

If $d_R = 0$, we have $R = K$ and $Ab_R = Ab_K \neq 0$ based on the previous lemma. We now assume that for any enclosing convex polygon R whose $d_R = N - 1$ we have $Ab_R = Ab_K \neq 0$. Given a region R such that $d_R = N$, we will consider the shortest side i of ∂R such that $d_{R,i} > 0$. Given that it is the shortest, $b_{R,i-1}$ and $b_{R,i+1}$ must both be positive or we will have two zero-length sides ($b_{R,i} = 0$ and either $b_{R,i_1} = 0$ or $b_{R,i+1} = 0$) which corresponds to a π turn, i.e., a degenerate case. Consider the straight path γ that is parallel to side i and has a distance of $d_{R,i} - 1$ to K (Figure 15). Then γ must intersect sides $i-1$ and $i+1$ of R at an angle of $\frac{\pi}{3}$ since $b_{R,i-1} > 0$ and $b_{R,i+1} > 0$. Consequently, removing from R the trapezoidal strip bounded by γ and sides $i-1$, i, and $i+1$ of R results in a convex hexagon R'. Number the sides of R' such that its side i is on γ and its sides $i-1$ and $i+1$ are on the sides of $i-1$ and $i+1$ of R, respectively. From the construction of R' we have $d_{R',i-1} = d_{R,i-1}$, $d_{R',i+1} = d_{R,i+1}$, and $d_{R',i} = d_{R,i} - 1$. Consequently, $d_{R'} = d_R - 1 = N - 1$ and by assumption $Ab_{R'} = Ab_K \neq 0$. Moreover, $b_{R',i-1} = b_{R,i} - 1$, $b_{R',i+1} = b_{R,i+1} - 1$, and $b_{R',i} = b_{R,i} + 1$ while other sides are unaffected. Consequently, $b_{R,i-1} + b_{R,i} = b_{R',i-1} + R_{R',i}$, $b_{R,i+1} + b_{R,i} = b_{R',i+1} + R_{R',i}$, and $b_{R,i-2} + b_{R,i-1} = b_{R',i-2} + b_{R',i-1} - 1 = b_{R',i+2} + b_{R',i+1} - 1 = b_{R,i+2} + b_{R,i+1}$. In other words, $Ab_R = Ab_{R'} = Ab_K \neq 0$. ∎