

# Semi-regular Quadrilateral-only Remeshing from Simplified Base Domains

Joel Daniels II<sup>1</sup>, Claudio T. Silva<sup>1,2</sup> and Elaine Cohen<sup>1</sup>

<sup>1</sup> School of Computing, University of Utah, United States

<sup>2</sup> Scientific Computing and Imaging Institute, University of Utah, United States

---

## Abstract

*Semi-regular meshes describe surface models that exhibit a structural regularity that facilitates many geometric processing algorithms. We introduce a technique to construct semi-regular, quad-only meshes from input surface meshes of arbitrary polygonal type and genus. The algorithm generates a quad-only model through subdivision of the input polygons, then simplifies to a base domain that is homeomorphic to the original mesh. During the simplification, a novel hierarchical mapping method, keyframe mapping, stores specific levels-of-detail to guide the mapping of the original vertices to the base domain. The algorithm implements a scheme for refinement with adaptive resampling of the base domain and backward projects to the original surface. As a byproduct of the remeshing scheme, a surface parameterization is associated with the remesh vertices to facilitate subsequent geometric processing, i.e. texture mapping, subdivision surfaces and spline-based modeling.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid and object representations

---

## 1. Introduction

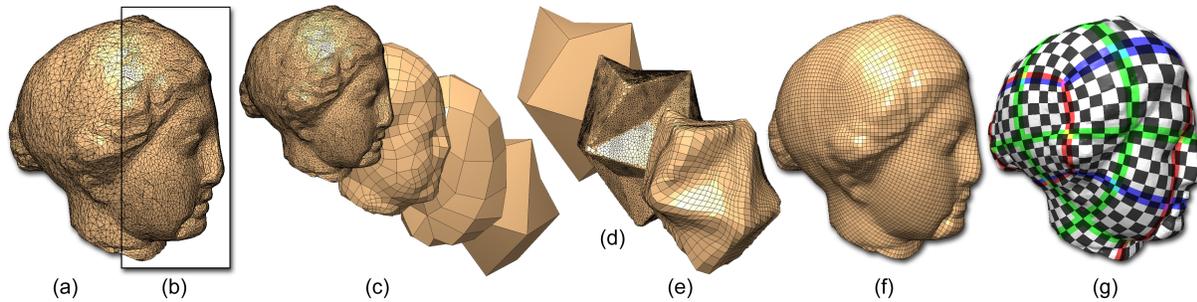
Polygonal models can be categorized as *irregular* or (*semi*) *regular* determined by structural properties of the mesh connectivity. An irregular mesh sacrifices strict connectivity-based constraints for a degree of freedom to better accommodate the description of complex geometric features, model deformations and tracking topological changes. In contrast, a regular mesh requires an exact vertex valence maintained by all internal vertices of the model.

For a quad mesh, a completely regular mesh is defined to be one where all vertices have valence 4. This constraint is difficult, often impossible, to satisfy, as only genus-1 (toroidal) models can be described as a regular quad mesh. However, despite the burdensome connectivity-based constraints, the regularity of the mesh structure facilitates processing algorithms. Consequently, surface parameterizations including texturing [THCM04] and spline-based modeling [WHL\*07], mesh subdivision [CC78], Fourier- [PSZ01] and wavelet-based [UCB04] computations, mesh compression [KSS00] and comparison [PSS01] algorithms exploit assumptions about the neighborhood connectivity.

A semi-regular model relaxes the structural constraints by allowing some number of extraordinary (non-ideal valence) vertices that define the boundary curves of a coarse segmentation of the model. Internally, each of the coarse regions is described by a regular mesh structure. Semi-regular meshes are able to describe surface models of arbitrary genus, while exhibiting the structural regularity that facilitates many geometric processing algorithms as illustrated in Fig. 1.

We address the generation of semi-regular quad-only meshes, because the extraordinary vertices of these meshes define a coarse quad segmentation of the model. A quad element shares a common domain with surface parameterization solutions, i.e. texture mapping and spline-based modeling. In this way, a coarse quad-only segmentation facilitates geometric processing, and, as a byproduct of our algorithm, generates a parameterization over the original surface.

This work uses quad-based simplification to build the base domains. It is well articulated in related simplification research [DSSC08, SBS08] that the quad element enforces structural constraints on the mesh, where the deletion of a single quad may require the removal of a larger collection of



**Figure 1:** Our algorithm splits an input mesh of arbitrary polygonal type (a) into a quad-only mesh (b), simplifies the model while maintaining critical levels-of-detail (c) to guide the map of the original geometry to the base domain (d). The base domain is refined, the vertices relaxed to accommodate for area distortions in the map (e), then the vertices are reprojected to the original surface (f). A surface parameterization is a byproduct of the method to facilitate geometry processing, i.e. texture mapping (g).

elements to preserve the structure of an all-quad mesh. A key novelty of this research is a mapping technique that is not dependent on the particular coarsening operations.

While semi-regular, quad-only meshes demonstrate structural advantages useful in subsequent applications, their constructions are complicated by parameterization-based challenges. The base domain coarseness is constrained by the genus and geometric complexities of the model, where too few base elements typically lead to distorted elements and poor surface approximation, illustrated in Fig. 7. To address these problems, we propose to adaptively sample the base domain, using an approximation to the surface area as well as element quality to reduce the parametric distortion and improve remeshing errors.

**Contributions.** In this paper, we propose an algorithm to remesh input polygonal-based surfaces of arbitrary genus. The algorithm refines a base domain mesh simplified from the input model, a mapping of the original vertices onto the base quads allows for a backward projection of the remesh vertices. The key contributions of this paper are: (1) a hierarchical mapping technique that supports arbitrary, i.e. local and global, deletion operators while supporting simplification to *very* coarse base domains (i.e., 10 quads); (2) an adaptive resampling of the base domain to reduce parametric distortions; (3) a semi-regular, quad-only remeshing approach that can be generalized to arbitrary polygonal remeshing.

## 2. Related Research

Quad meshes are increasing in popularity. Because of the added complexity and structural constraints of quad meshes, many quad-processing algorithms generate irregular meshes or quad-dominant representations. For instance, triangle merging [MK04, LKH08] and advancing front algorithms [OSCS99] that facilitate the conversion of non-quad elements robustly generate quad-dominant models. Resampling techniques [VSI00], using rectangularly packed repul-

sion forces, obtain a good distribution of points over the model to describe quad-dominant models without low element quality related to the front collisions. Surface grafting [JBSM99, BPJH02] uses inside/outside tests over a volumetric voxelization to guide irregular quad-only mesh generation. Numerical integration of orthogonal vector fields, tracing harmonic function gradients [KNP07, DKG05] and principal curvature directions [ACSD\*03, MK06], yields high quality quad-dominant meshes.

More rigorously constrained semi-regular, quad-only remeshes lend themselves to parameterizing surfaces. Best categorized as divide-and-conquer algorithms, these methods segment the model into a set of coarse quad regions that are individually remeshed. Although an early approach relied on user-guided graph cuts [KL96], more recent techniques have developed automatic segmentations using harmonic functions [NGH04] and orthogonal vector fields [RLL\*06]. The evaluation of frequency-related eigenfunctions of the mesh's Laplacian matrix [DBG\*06, TACSD06] describes a coarse quad segmentation of the original geometry. An extension of the spectral quadrangulation, using additional weighting matrices, seeks attribute alignment and adaptive mesh sampling [HZM\*08].

Other semi-regular, quad remeshing algorithms attempt to place extraordinary vertices at regions of high curvature. For instance, normal based clustering used to guide a coarse quad segmentation exhibits alignment of the base domain to curvature directions [BMRJ04]. Similarly, user driven coarse vector fields that resemble the low frequency surface geometry improve anisotropic remeshing with a limited number of extraordinary vertices [BZK09].

In contrast to these remeshing algorithms, we propose a simplification-based technique that relies on robust, connectivity-based operations in place of numerical integration. This approach is flexible and straightforward to generalize to arbitrary polygonal remeshes, and has the potential to be useful in volumetric modeling as these methods are re-

lated to manual operators developed in hex-based research. Our algorithm allows the user to directly and intuitively control the number of base domain quads and extraordinary vertices in the remesh, while new vertices are automatically located to improve element quality and approximation errors.

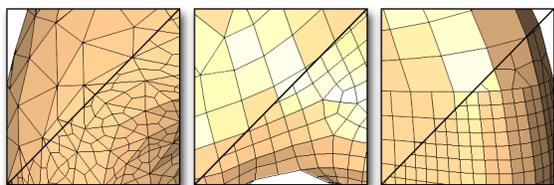
**Quad-based Simplification.** Simplification deletes selected primitives from the model to reduce the number of defining elements until breaching prescribed tolerance thresholds. The deletion ordering is generally determined by the element's importance to the surface description, successfully encoded in triangle-based schemes using a quadric error metric [GH97]. Other research investigates the inclusion of additional metrics, including element quality and vertex valence [SBM05] or appearance attributes [Hop99].

Maintaining quality elements during simplification is an important aspect of many quad-based simplification schemes. Some improvement schemes and simplification algorithms [SC97, Kin97] study the effects of localized deletion operators on mesh structure. However, it has been well articulated in previous work [BPJH02, SBS08, DSSC08] that globalized deletion operators are critical in maintaining high quality mesh structures, further discussed in Sec. 3.1.

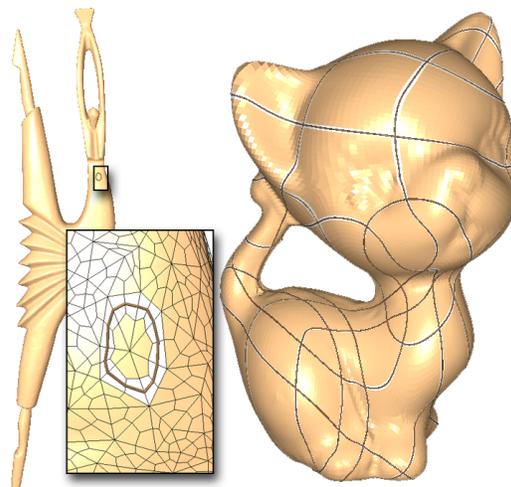
**Simplification-based Remeshing.** Triangle remeshing schemes leveraging simplification and refinement [KLS03, AGL06], in particular, the MAPS [LSS\*98] technique inspired our work. An input model is simplified to a desired base domain maintaining a conformal mapping of the original connectivity on each intermediate level-of-detail. Regular refinement of the base mesh, combined with backward projections based on the conformal mapped mesh data, yields semi-regular triangle-based representations. We build on these principles, proposing a simplification-based scheme for quad remeshing that works with arbitrary deletion methods, can be generalized for arbitrary polygonal remeshes, and adaptively resamples the base domain to accommodate for parametric distortions especially when simplifying to very coarse models.

### 3. Semi-regular Remeshing

The semi-regular, quad-only remesh is constructed from input polygonal-based meshes. This work stems from the observation that a single execution of a splitting scheme based



**Figure 2:** Splitting the polygonal mesh based on Catmull-Clark subdivision rules yields quad-only elements.



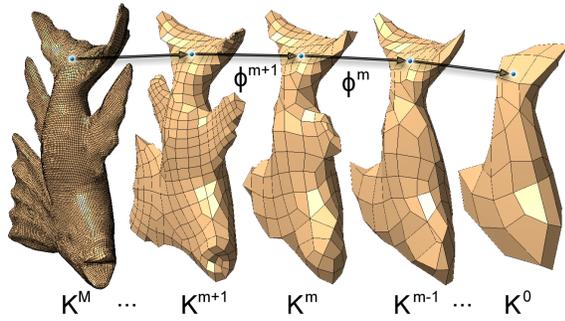
**Figure 3:** A single polychord is highlighted on each model. While mapping some polychord neighborhood boundaries to the plane is straightforward (left), the global nature of these structures may necessitate other parametric domains (right).

on Catmull-Clark subdivision [CC78], results in quad-only representation of the input model independent of the original polygonal elements. For instance, illustrated in Fig. 2, Catmull-Clark subdivision yields quad-only reconstructions of triangle and quad-dominant models, as well as quad meshes with T-junctions. This iteration inserts ideal vertices (valence 4) at the midpoints of the mesh edges, and vertices with valence equal to the polygon sides at each face centroid. Following subdivision, the algorithm executes the simplification and refinement operations on these quad-only representations to generate the remesh.

### 3.1. Deletion Operators

Quad-based simplification constructs the base domain mesh, while maintaining a mapping from the original model to the coarsened mesh at each levels-of-detail. In contrast to triangle-based techniques [LSS\*98], mapping quad-based simplifications have the challenge of supporting a large cast of deletion templates, including global operators. In particular, as discussed in previous simplification research [DSSC08, SBS08], the *dual polychord collapse operator* is a critical deletion operator for quad meshes.

The derived dual representation [BBS02] of quad meshes aids in analysis and processing. It is defined to have the following components: the dual of a quad is its *centroid*; the dual of an edge is its *chord* that connects the two centroids of the neighboring quads; and the dual of a vertex is its *polychord* that connects the centroids of the neighboring quads in a cyclic order. The polychord is a higher-order dual structure, defined as a polyline whose adjacent segments are chords



**Figure 4:** Keyframe meshes  $K^m$  are discrete samplings of the simplification hierarchy, used to guide the mapping of a point from  $K^M$  to  $K^0$ .

that meet at a common centroid and are dual to opposing edges in that quad.

Deleting a polychord merges the vertex endpoints of all edges to which it is dual, simultaneously removing multiple quads from the model. While mapping some polychord neighborhoods to the plane may be straightforward, the complex knots and global nature of these structures can quickly complicate the parameterization method (Fig. 3). In this work, we use a variation of the QMS simplification algorithm [DSSC08]. This techniques describe various simplification operators, global and local, as well as weighting functions for the automated prioritization of element deletions. To support the differing deletion types without special case handling, we propose a novel hierarchical mapping scheme.

### 3.2. Keyframe Meshes

The function  $\phi$  defines a bijective mapping of the vertices  $V$  of an input quad mesh  $M$  to the base domain mesh  $M^0$ ,  $\phi : M \rightarrow M^0$ . To support arbitrary deletion operators without special case handling,  $\phi$  is constructed by storing a set of keyframe meshes  $K^{\{M, \dots, 0\}}$  during the simplification process, illustrated in Fig. 4. The term keyframe is intended to evoke a popular animation technique, where important locations and poses are defined through which a character deforms. Analogously, the keyframe meshes dictate the path progression for points as they map from  $M$  to  $M^0$ .

The original model  $M$  is pushed onto the stack of keyframe meshes, denoted as  $K^M$ . During simplification a new keyframe mesh  $K^{m-1}$  is committed to stack as necessitated by an inspection routine executed after each deletion iteration. The current simplified mesh  $M^c$  is committed to the keyframe stack if the Hausdorff distance between  $M^c$  and  $K^m$ , the previously committed keyframe mesh, is greater than a specified distance  $d$ ; or if the projection of  $K^m$  onto  $M^c$  has flipped elements. Lastly, the base domain mesh  $M^0$  is committed as the final keyframe mesh  $K^0$ .

To improve performance, this inspection process is local-

ized to a subset of quads  $Q^m$  of  $K^m$ . Consider that a deletion operator processes a set quads  $Q^c$  of  $M^c$ , including the element(s) intended for deletion and their one-neighborhood, returning a new set of quads  $Q^{new}$ , where  $|Q^{new}| < |Q^c|$ . Only the subset of quads  $Q^m$  of  $K^m$  within the distance  $d$  of the original quad group  $Q^c$  are considered. The vertices of  $Q^m$  are projected onto  $Q^{new}$  using a closest point projection, testing the distance threshold and for flipped elements. Localizing the inspection improves performance.

### 3.3. Hierarchical Keyframe Mapping

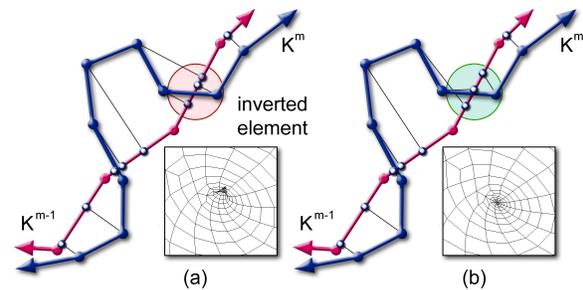
The development of  $\phi : M \rightarrow M^0$  is guided by the keyframe meshes, where individual functions are independently developed to map each keyframe mesh to the next coarsest representation,  $\phi^m : K^m \rightarrow K^{m-1}$ . As illustrated in Fig. 5, the function  $\phi^m$  is obtained through iterative ray-casting and relaxation of the vertices of  $K^m$  over  $K^{m-1}$  until inverted elements are resolved. Fig. 5 illustrates a 2D diagram of the projection and relaxation phase results, as well as an example image of the fold-over evident within a projected mesh.

A new mesh  $\tilde{K}^m$  that is the projection of  $K^m$  onto  $K^{m-1}$ , produces an initial  $\phi^m$ . To ensure that  $\phi^m$  is a bijective mapping, flipped elements in  $\tilde{K}^m$  are resolved via a relaxation phase. A movement vector  $\tilde{m}$  corresponding to a vertex  $\tilde{v}$  of  $\tilde{K}^m$  is computed towards the weighted average of the centroids of the neighboring quads  $\tilde{q}_i$  of  $\tilde{v}$ :

$$\tilde{m} = \frac{\sum_i C(\tilde{q}_i) \theta(\tilde{N}^m(C(\tilde{q}_i)), N^{m-1}(\tilde{v}))}{\sum_i \theta(\tilde{N}^m(C(\tilde{q}_i)), N^{m-1}(\tilde{v}))} - \tilde{v},$$

$$\theta(n_1, n_2) = \begin{cases} 10, & \langle n_1, n_2 \rangle \geq 0.0 \\ 0, & \text{otherwise} \end{cases}$$

where  $C(q)$  computes the centroid of quad  $q$ ,  $\tilde{N}^m(v)$  returns the normal of  $\tilde{K}^m$  evaluated at  $v$ ,  $N^{m-1}(v)$  returns the normal of  $K^{m-1}$  evaluated at the projection of  $v$ , and  $\langle n_1, n_2 \rangle$  is the inner product of the two vectors. The relaxation process resolves flipped elements of  $\tilde{K}^m$  by assigning larger weights,  $\theta$ , to non-flipped quads. The weighting differential results



**Figure 5:** The function  $\phi^m$  maps the vertices and connectivity of the keyframe mesh  $K^m$  onto the next keyframe mesh  $K^{m-1}$ , developed as a two phase process: ray-cast projection (a) and relaxation to resolve inverted elements in the projection of  $K^m$  (b).

in a pulling effect that spreads points away from flipped regions. The movement vector  $\tilde{m}$  is projected and scaled,  $\tilde{v} = \tilde{v} + \alpha(\tilde{m} - \langle N^{m-1}(\tilde{v}), \tilde{m} \rangle N^{m-1}(\tilde{v}))$ , in practice  $\alpha = 0.5$ , and  $\tilde{v}$  is reprojected onto  $K^{m-1}$ .

Developing the mapping functions  $\phi^m$  between keyframe meshes allows the procedure to be parallelized with the simplification process and each other. This hierarchical mapping technique improves computational performance required in the resolution of flipped elements, because each keyframe mesh stores a reduced number of vertices, especially in comparison to  $M$ . The projection of a point on  $M$  to  $M^0$  through the keyframe mapping functions  $\phi^m$  necessitates the use of barycentric coordinates, illustrated in Fig. 4. In this work, the barycentric coordinates are computed by virtually dividing each quad into four triangles, radiating about the centroid.

### 3.4. Downward Projection

A point  $p$  on  $K^m$  is assigned the barycentric coordinates  $(\alpha, \beta, \gamma)$  for the sub-triangle  $t$  of the quad  $q$ . The vertices of  $q$  are indexed  $q.v_i$ ,  $i = (0, 1, 2, 3)$ , the centroid is  $q.c$ , and the sub-triangle  $t$  is described by vertices  $(q.v_i, q.v_{(i+1)\%4}, q.c)$ . The projection of  $p$  onto  $K^{m-1}$  is computed based on  $\phi^m$ , illustrated in Fig. 6.

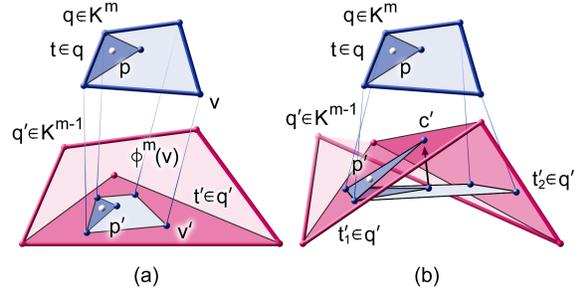
When the vertices of  $q$  map to the same sub-triangle  $t'$  of quad  $q'$  on  $K^{m-1}$ , mapping  $p$  onto  $K^{m-1}$  is straightforward (Fig. 6a). Barycentric coordinates are computed for each vertex  $q.v_i$  on  $t'$  as  $(\alpha_i, \beta_i, \gamma_i)$ , and those assigned the centroid are an average of the four vertices,  $(\alpha_c, \beta_c, \gamma_c) = \frac{1}{4} \sum_{i=0}^3 (\alpha_i, \beta_i, \gamma_i)$ . The new barycentric coordinates for  $p$  within  $t'$  are computed as a weighted combination,

$$\begin{aligned} \alpha'_t \cdot \alpha + \alpha'_{(t+1)\%4} \cdot \beta + \alpha'_c \cdot \gamma, \\ \beta'_t \cdot \alpha + \beta'_{(t+1)\%4} \cdot \beta + \beta'_c \cdot \gamma, \\ \gamma'_t \cdot \alpha + \gamma'_{(t+1)\%4} \cdot \beta + \gamma'_c \cdot \gamma. \end{aligned}$$

The more challenging problem is when the vertices of  $q$  map to multiple sub-triangles on  $K^{m-1}$ . When the vertices of  $q$  map to two adjacent sub-triangles,  $t'_1$  and  $t'_2$ , on  $K^{m-1}$ , the triangles may be flattened by unhinging the edge between them. On this plane, new barycentric coordinates may be computed for  $p$  after projecting the vertices  $q.v_i$  and  $q.c$ . However, when additional sub-triangles are involved, more intricate flattening strategies are required.

Instead, to compute new barycentric coordinates of  $p$ , we decided to use a ray-casting approach (Fig. 6b). The vertices  $q.v_i$  correspond to  $q.v'_i$  of  $K^m$ ,  $q.v'_i = \phi^m(q.v_i)$ . Because the vertices of  $q$  map to multiple sub-triangles of  $K^{m-1}$ , the simple projection case (Fig. 6a) does not apply. Instead, the projected centroid  $q.c'$  is the average of the mapped vertices  $q.c' = \frac{1}{4} (\sum_i q.v'_i)$ . This point  $q.c'$  is projected in the normal direction  $\tilde{N}^m(q.c')$  onto  $K^{m-1}$ .

If the vertices  $q.v'_i$ ,  $q.v'_{(i+1)\%4}$ , and  $q.c'$ , map to the same sub-triangle of  $K^{m-1}$ , then new barycentric coordinates for



**Figure 6:** The barycentric coordinates of the point  $p$  within sub-triangle  $t$  of  $q \in K^m$  are known. If all vertices of  $q$  map,  $\phi^m(v)$ , to the same sub-triangle (a)  $t' \in K^{m-1}$ , then new barycentric coordinates assigned to  $p$  are computed at  $p'$  within  $t'$ . If the vertices of  $q$  map to multiple sub-triangles (b), the mapped centroid is projected in a normal direction to  $K^{m-1}$ ,  $c'$ , and  $p'$  is computed. If  $p'$  is not on a sub-triangle of  $K^{m-1}$ , then it is projected in a normal direction.

$p$  may be computed as a weighted combination, previously discussed. However, when these vertices map to multiple sub-triangles,  $p'$  is computed on the triangle formed by these vertices,  $p' = \alpha q.v'_i + \beta q.v'_{(i+1)\%4} + \gamma q.c'$ . This point is projected in the normal direction  $\tilde{N}^m(p')$  onto  $K^{m-1}$ , computing new barycentric coordinates for  $p$  at the intersection.

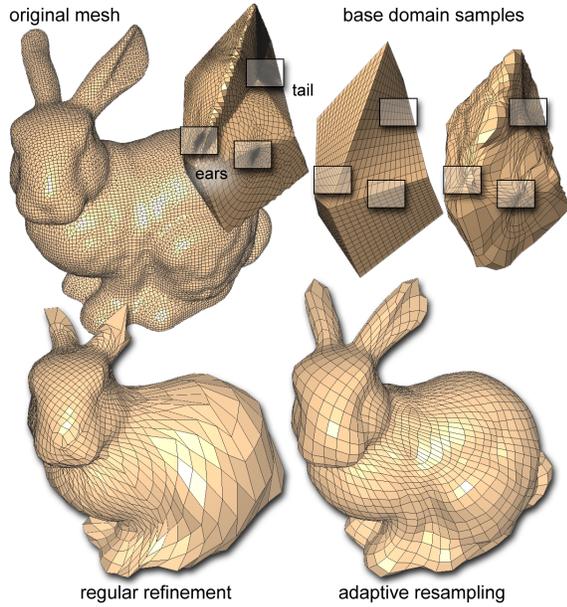
The ray-casting based downward projections yield similar results to the previously described unhinging technique, without special cases for the various neighborhood scenarios. For improved performance, the vertices and centroids of each keyframe mesh are projected once and stored. Future projections require only normal projections of  $p'$  for a subset of the sub-triangles in  $K^m$ .

### 3.5. Adaptive Resampling

Following the computation of a map, typically a semi-regular remesh is computed through regular refinement of  $K^0$  and backward projection of the vertices onto  $K^M$ . However, this approach is unable to accommodate for non-equiareal mappings that results in poor surface approximations. We allow the base domain remesh  $R^0$  to adaptively resample  $K^0$  (Fig. 7), guided by surface area, approximation error and the element quality of the final remesh  $R^M$ .

The vertices of  $R^0$  lie on the base domain determined by regular refinement of  $M^0$ . The area of the original model  $M$  associated with each vertex  $v \in R^0$  can be computed by integrating the area of  $M$  that maps onto  $M^0$  nearest to  $v$ . Relaxation of  $v$  occurs by moving towards the area-weighted centroid of its neighboring remesh vertices. Iterative execution of the relaxation improves the distribution of the remesh vertices, more evenly sampling the original model.

Because our keyframe mapping approach does not describe a conformal mapping, the angles formed by edges



**Figure 7:** Regular refinement of the base domain may poorly approximate the surface, due to area-based distortions in the mapping. Our refinement and adaptive resampling better accommodates these regions, highlighted on the tail and ears.

of  $R^0$  do not translate to similar angles on  $R^M$ . A second relaxation phase is integrated within the resampling to reduce parametric distortion on  $R^M$  by adapting a *ballooning scheme* that improves element quality and approximation [SLS\*06]. Each vertex in  $R^M$  moves in the direction of the vertex normal, scaled by the accumulated error value measured as the signed distance between each neighboring quad centroid and  $M$ . The vertices are simultaneously relaxed toward the the average of their connected neighbors, and projected onto  $K^M$ . The vertices of  $R^0$ ,  $v_i^0$  corresponding to  $v_i^M$  on  $R^M$ , are updated to reflect these relaxations,  $v_i^0 = \phi(v_i^M)$ .

This process leverages both representations of the remesh on the base domain  $R^0$  and the original model  $R^M$ . The relaxations improve element quality while allowing the remesh to cope with parametric distortions in the map, illustrated in Fig. 7. Furthermore, hierarchical resampling, achieved by interleaving the refinement and two relaxation phases produces faster convergence of the remesh vertices, a method used for the remeshes illustrated throughout this paper.

**Area Approximation.** Wavefront propagation used to compute the surface area associated with each remesh vertex, as described above, is time consuming and costly. Instead, to quickly approximate the area of the original model as it maps to the base domain based on the keyframe maps, we construct two point clouds, using a kd-tree to facilitate nearest neighbor searches:  $P^M$  is a near equi-areal, random sampling of  $K^M$ , and  $P^0$  is its mapping onto  $K^0$  through the keyframe

maps,  $\phi^m$ . Given  $P^0$ , the approximation of the surface area associated with a remesh vertex  $v$  of  $R^0$  is computed by summing the number of points in  $P^0$  within a specified distance of  $v$ . The search radius is evaluated as one-half the average distance between  $v$  and each of its neighbor vertices in  $R^0$ . Because the sampling of  $P^M$  is assumed to be near equi-areal, the neighborhood count serves as a sufficient scalar to approximate an area-based weight that can be assigned to each point in  $R^0$ .

**Projections.** It is possible to project all of the original vertices through  $\phi$  onto the base domain, illustrated in Fig 7, for precise backward projection computations. However, for improved computational performance, we leverage the correspondence between the points in  $P^M$  and the projected points  $P^0$ . An approximation of the backward projection for a point  $p$  on  $K^0$  with neighboring points  $n_i \in P^0$  that correspond to  $n'_i \in P^M$ , is computed as

$$p' = \left( \sum_i \frac{n'_i}{\|p - n_i\|} \right) / \left( \sum_i \frac{1}{\|p - n_i\|} \right).$$

With a dense sampling of  $P^M$  (in practice  $2k$  points), the technique is fast and adequate for our purposes, avoiding the mapping of potentially many vertices in  $M$  to  $K^0$ . The subsequent ballooning, relaxation and projection will ensure that  $p'$  is placed on  $M$ . Furthermore, the relationship between  $P^M$  and  $P^0$  can be further exploited during the downward projections while updating the point locations of  $R^0$  to reflect relaxations that occur on  $R^M$ .

**Feature Preservation.** The simplification operators maintain feature edge loops, annotated as important structures on the original mesh [DSSC08]. The keyframe mapping respects feature by sampling the coarse features evenly with the feature vertices of  $K^m$  during the ray-casting phase and fixing these locations during the subsequent smoothing. During the adaptive resampling method, annotated feature points, corresponding to those sampled along the base domain feature edges, are not allowed to move. In this way, feature points may be adaptively sampled along the feature curves of  $M^0$  and faithfully backward projected to  $M$ .

## 4. Results

The quadrilateral mapping described in Sec. 3 was implemented in C++. The remeshes were performed on a 2.16 GHz Intel Core 2 Duo with 2GB memory, taking in the order of a few minutes to compute, further detailed in Table 1 for remeshes shown throughout the paper. The timings measure four phases, monitoring the simplification of the model (I), the additional time needed to complete the keyframe mappings (II), and the adaptive resampling and backward projections (III). This code had been built to emphasize its ability to operate independent of the simplification technique, supporting any variety of deletion operations without special case handling.

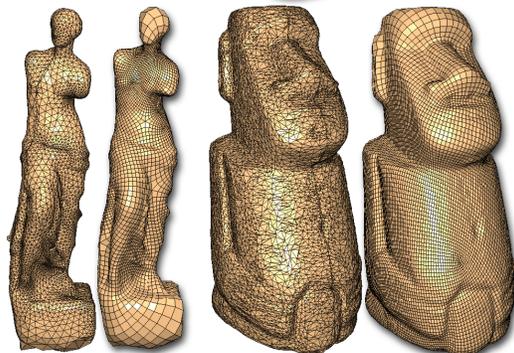
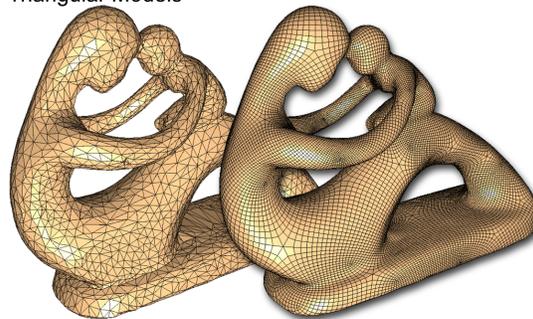
The implementation is tested on a range of models with varying genus, geometric complexities, and input polygonal types, illustrated in Fig. 8. These remeshes test multiple quad-based simplification algorithms that support locally- and globally-based operations, while developing quad-only reconstructions of triangle-, quad-dominant and irregular quad-only meshes. All of the remeshes shown throughout this paper emphasize the advantages of our simplification-based algorithm by supporting very coarse base domains.

Table 1 quantitatively analyzes the quality of the remeshes, documenting approximation errors, number of extraordinary vertices (non-ideal valence 4), worst case valence, and the orthogonality of the resulting parameterization. The remesh error measures the Hausdorff distance of the remesh and the original model, relative to the bounding box diagonal  $d_B$ . The number of extraordinary vertices is related to the number of coarse quad regions that segment the model for parameterization or semi-regular remeshing. The scaled Jacobian statistics indicate the orthogonality of the mesh elements, average and worst case, where 1 corresponds to a rectangle, 0 to a quad with 3 co-linear vertices, and inverted elements are less than 0.

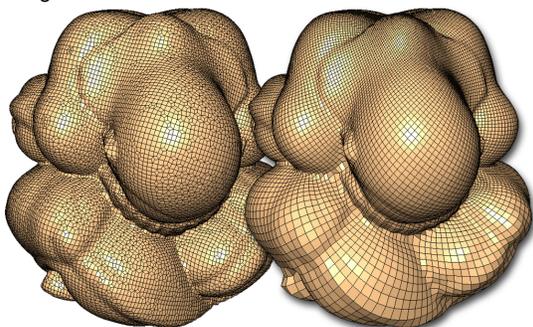
Further analysis compares the results of the adaptive remeshing versus traditional regular refinement, illustrated in Fig. 7. Regular refinement, especially when simplifying to coarse base domains, yields higher approximation error ( $E = 0.16d_B$ ) than our adaptive technique ( $E = 0.043d_B$ ). Without incorporating the relaxation scheme, regular refinement does not handle parametric distortions, generating a median scaled Jacobian of 0.81 with a worst case  $-0.94$ . Our adaptive resampling relaxes vertex locations based on element quality to improve these metrics, with a median scaled Jacobian equal to 0.88 and a worst case  $-0.25$ . The coarseness of the base domain (10 faces) can result in the negative scaled Jacobians, despite our adaptive resampling technique. In these cases, the small number of user desired extraordinary vertices over constrains the structure and optimization (Table 1). Allowing more extraordinary vertices and concomitantly more faces in the base domain gives the flexibility needed to improve the remesh quality.

**Remesh Comparison.** This study measures the quality of the remesh elements, a statistical analysis of the scaled Jacobians and the mesh angles, as well as a comparison of parametric stretching related to the mesh edge lengths. Our bimba remesh is compared to a model acquired online, remeshed using *periodic global parameterization* (PGP) [RLL\*06] in Fig. 5. The PGP remesh describes a coarse quad segmentation, generating a semi-regular, quad-only mesh. The 915 T-junctions on the model were *not* included in the count of extraordinary vertices. In comparison to PGP, our model, that was specifically constructed to have a similar number of extraordinary vertices, exhibits a similar statistical analysis (mean and standard deviation) of the mesh angles and edge lengths, while improving the worst case scaled

Triangular Models



Irregular Quad Models



**Figure 8:** Semi-regular, quad-only remesh results of our algorithm, supporting both local- and global-based simplification algorithms, for input triangle and quad-only models.

Jacobian. This comparison, and similar quality metrics evaluated on our remeshes (Table 1), illustrates that it is possible to create remeshes of similar quality with our method in comparison to existing remeshing methods, while improving the ability to control the number of base patches and extraordinary vertices of the final remesh.

**Applications.** Geometric processing algorithms are able to take advantage of the neighborhood structure offered by semi-regular polygonal meshes. Particularly, simplification-based remeshing schemes facilitate mesh improvement [BPJH02], consistent remeshing [SAPH04] and deformation [LDSS99] applications. As illustrated in Fig. 1, our semi-regular, quad-only remeshes are coupled with a sur-

Model	Original Vertices (IVl, lExl, Worst)	Time (seconds)		Remesh Vertices (IVl, lExl, Worst)	Scaled Jacobians (Median, Worst)	Remesh Angles (Median, $\sigma$ )	Error ( $10^{-2}$ )
		(I,II,III)	T				
Egea (Fig. 1)	(27k, 13.1k, 11)	(26, 30, 28)	84	(10.2k, 8, 5)	(0.98, 0.66)	(89.9°, 9.6°)	0.71 $d_B$
Bunny (Fig. 7)	(21.7k, 9.6k, 6)	(34, 1348, 33)	1415	(2.3k, 8, 5)	(0.88, -0.25)	(89.3°, 32.5°)	4.3 $d_B$
Fertility (Fig. 8)	(22.5k, 11.1k, 11)	(18, 9, 42)	69	(31.9k, 110, 6)	(0.98, 0.06)	(89.9°, 9.7°)	1.0 $d_B$
Venus (Fig. 8)	(28.1k, 14k, 11)	(139, 15, 39)	193	(25.6k, 26, 6)	(0.98, 0.45)	(89.8°, 10.7°)	1.4 $d_B$
Moai (Fig. 8)	(63.5k, 31k, 11)	(293, 7, 24)	324	(12.8k, 12, 5)	(0.98, 0.45)	(89.7°, 10.7°)	0.083 $d_B$
Pensatore (Fig. 8)	(31.1k, 16.4k, 6)	(66, 31, 33)	130	(19.5k, 8, 5)	(0.99, 0.41)	(89.8°, 8.6°)	0.16 $d_B$

**Table 1:** Analysis of the remesh times (simplification (I), keyframe mapping (II), adaptive remesh (III) and total (T)), vertex information (total, extraordinary, and worst case valence), element quality (median and worst Scaled Jacobian, median and standard deviation mesh angles), and approximation errors of the models shown throughout the paper.

face parameterization as a byproduct of their construction. Texturing and displacement mapping applications, as well as spline-based modeling is straightforward.

**Limitations.** The hierarchical mapping technique can generalize to other surface representations, supports arbitrary deletion methods, and describes a hierarchical approach to the map development. However, the relaxation that resolves inverted projections can require many iterations, especially in resolving large fold-over regions that may occur while reducing to a very coarse base domain. Most remeshes are obtained within a few minutes (Table 1); however, the Stanford bunny (Fig. 7) required 23 minutes because of the base domain coarseness. This mapping technique is unable to handle cases where the simplification generates self-intersections.

Future research will address improving the placement of the remesh vertices, in particular, extraordinary vertices. An advantage of our approach is that it will facilitate processing, by computing vertex shifting, element refinement, and other methods on the coarse base domains. An important and challenging aspect of quad meshes is to address the placement of base domain extraordinary vertices and the integration of attribute alignment [LKH08].

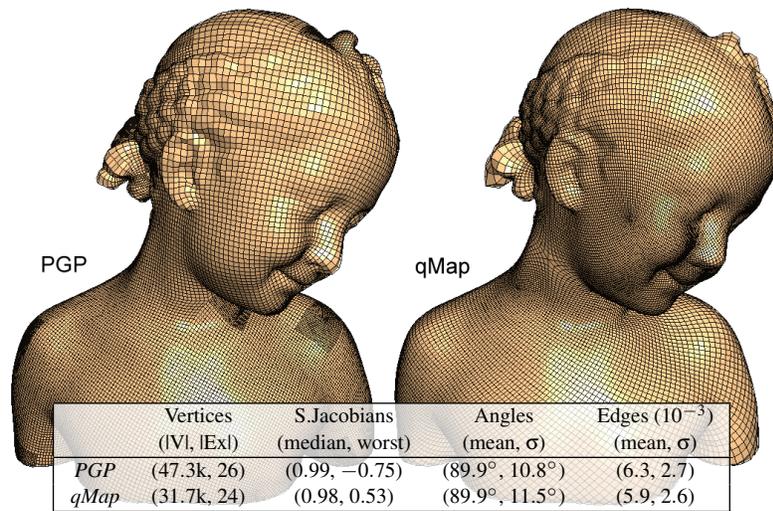
## 5. Conclusion

We introduce a simplification-based technique for the semi-regular, quad-only remeshing of arbitrary topological polygonal meshes that operates independent of the deletion operations by leveraging *keyframe* meshes to guide a hierarchical mapping algorithm. It is shown that our method can produce models with similar quality elements as an existing quad remeshing scheme (PGP), while providing tools for more direct control over the number of extraordinary vertices to produce *very* coarse quad regions. The remesh vertices are sampled in a way that reduces parametric distortions and approximation errors. The remeshing algorithm is able to significantly simplify the input geometry by implementing an adaptive resampling scheme of the base domain to accommodate for area distortions in the mapping functions. The modified resampling supports more coarse segmentations than other simplification-based remeshing [DSSC08] and mapping-based methods [LSS\*98, KLS03, AGL06], by which this work is inspired.

**Acknowledgments.** We would like to the anonymous reviewers for constructive comments. We thank Jason Shepherd and the AIM@SHAPE project for access to 3D models used in our research. This research has been funded by NSF(CCF0541402, IIS0844546, ATM0835821, CNS0751152, CCF0528201, OCE0424602, CNS0514485, IIS0513692, CCF0401498, OISE0405402, CNS0551724), DoE, and IBM Faculty Awards.

## References

- [ACSD\*03] ALLIEZ P., COHEN-STEINER D., DEVILLERS O., LÉVY B., DESBRUN M.: Anisotropic polygonal remeshing. In *ACM SIGGRAPH* (2003).
- [AGL06] AHN M., GUSKOV I., LEE S.: Out-of-core remeshing of large polygonal meshes. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006).
- [BBS02] BORDEN M., BENZLEY S., SHEPHERD J.: Hexahedral sheet extraction. In *11th International Meshing Roundtable* (September 2002).
- [BMRJ04] BOIER-MARTIN I., RUSHMEIER H., JIN J.: Parameterization of triangle meshes over quadrilateral domains. *Symposium on Geometry Processing* (2004).
- [BPJH02] BREMER P., PORUMBESCU S., JOY K., HAMANN B.: Automatic semi-regular mesh construction from adaptive distance fields. *Curve and Surface Fitting: Saint-Malo* (2002).
- [BZK09] BOMMES D., ZIMMER H., KOBELT L.: Mixed-integer quadrangulation. *ACM SIGGRAPH* (2009).
- [CC78] CATMULL E., CLARK J.: Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer Aided Design* 10, 6 (1978).
- [DBG\*06] DONG S., BREMER P.-T., GARLAND M., PASCUCCI V., HART J. C.: Spectral surface quadrangulation. In *ACM SIGGRAPH* (2006).
- [DKG05] DONG S., KIRCHER S., GARLAND M.: Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Computer Aided Geometric Design* 22, 5 (2005).
- [DSSC08] DANIELS J., SILVA C., SHEPHERD J., COHEN E.: Quadrilateral mesh simplification. In *ACM SIGGRAPH Asia* (2008).
- [GH97] GARLAND M., HECKBERT P.: Surface simplification using quadric error metrics. In *ACM SIGGRAPH* (1997).
- [Hop99] HOPPE H.: New quadric metric for simplifying meshes with appearance attributes. In *IEEE Visualization* (1999).
- [HZM\*08] HUANG J., ZHANG M., MA J., LIU X., KOBELT L., BAO H.: Spectral quadrangulation with orientation and alignment control. In *ACM SIGGRAPH Asia* (2008).



**Figure 9:** Remesh results of periodic global parameterization (PGP) [RLL\*06] and our remeshing algorithm (qMap). A quantitative analysis comparing vertices (original, remesh, and extraordinary counts), and mesh angles and edge length statistics.

- [JBSM99] JANKOVICH S., BENZLEY S., SHEPHERD J., MITCHELL S.: The graft tool: An all-hexahedral transition algorithm for creating a multi-directional swept volume mesh. In *8th International Meshing Roundtable* (1999).
- [Kin97] KINNEY P.: Cleanup: Improving quadrilateral finite element meshes. In *6th International Meshing Roundtable* (1997).
- [KL96] KRISHNAMURTHY V., LEVOY M.: Fitting smooth surface to dense polygon meshes. In *ACM SIGGRAPH* (1996).
- [KLS03] KHODAKOVSKY A., LITKE N., SCHRODER P.: Globally smooth parameterizations with low distortion. *ACM Transactions on Graphics* 22, 3 (2003).
- [KNP07] KALBERER F., NIESER M., POLTHIER K.: Quadcover: Surface parameterization using branched coverings. *Computer Graphics Forum* 26, 3 (2007).
- [KSS00] KHODAKOVSKY A., SCHRODER P., SWELDENS W.: Progressive geometry compression. In *ACM SIGGRAPH* (2000).
- [LDSS99] LEE A. W., DOBKIN D., SWELDENS W., SCHRODER P.: Multiresolution mesh morphing. In *ACM SIGGRAPH* (1999).
- [LKH08] LAI Y.-K., KOBBELT L., HU S.-M.: An incremental approach to feature aligned quad dominant remeshing. In *ACM Solid and Physical Modeling Symposium* (2008).
- [LSS\*98] LEE A., SWELDENS W., SCHRODER P., COWSAR L., DOBKIN D.: Maps: Multiresolution adaptive parameterization of surfaces. In *ACM SIGGRAPH* (1998).
- [MK04] MARINOV M., KOBBELT L.: Direct anisotropic quad-dominant remeshing. In *Pacific Graphics* (October 2004).
- [MK06] MARINOV M., KOBBELT L.: A robust two-step procedure for quad-dominant remeshing. *Computer Graphics Forum* 25, 3 (2006).
- [NGH04] NI X., GARLAND M., HART J. C.: Fair morse functions for extracting the topological structure of a surface mesh. In *ACM SIGGRAPH* (2004).
- [OSCS99] OWEN S., STATEN M., CANANN S., SAIGAL S.: Q-morph: An indirect approach to advancing front quad meshing. *International Journal for Numerical Methods in Engineering* (March 1999).
- [PSS01] PRAUN E., SWELDENS W., SCHRODER P.: Consistent mesh parameterization. In *ACM SIGGRAPH* (2001).
- [PSZ01] PENG J., STRELA V., ZORIN D.: A simple algorithm for surface denoising. In *IEEE Visualization* (2001).
- [RLL\*06] RAY N., LI W. C., LEVY B., SHEFFER A., ALLIEZ P.: Periodic global parameterization. *ACM Transactions on Graphics* 25, 4 (2006).
- [SAPH04] SCHREINER J., ASIRVATHAM A., PRAUN E., HOPPE H.: Inter-surface mapping. In *ACM SIGGRAPH* (2004).
- [SBM05] SMITH J., BOIER-MARTIN I.: Combining metrics for mesh simplification and parameterization. In *ACM SIGGRAPH Sketches* (2005).
- [SBS08] STATEN M., BENZLEY S., SCOTT M.: A methodology for quadrilateral finite element mesh coarsening. *Engineering with Computers* (2008).
- [SC97] STATEN M. L., CANANN S. A.: Post refinement element shape improvement for quadrilateral meshes. *ASME AMD: Trends in Unstructured Mesh Generation* (1997).
- [SLS\*06] SHARF A., LEWINER T., SHAMIR A., KOBBELT L., COHEN-OR D.: Competing fronts for coarse-to-fine surface reconstruction. *Computer Graphics Forum* 25, 3 (2006).
- [TACSD06] TONG Y., ALLIEZ P., COHEN-STEINER D., DESBRUN M.: Designing quadrangulations with discrete harmonic forms. In *Symposium on Geometry Processing* (2006).
- [THCM04] TARINI M., HORMANN K., CIGNONI P., MONTANI C.: Polycube-maps. In *ACM SIGGRAPH* (2004).
- [UCB04] UCCHEDDU F., CORSINI M., BARNI M.: Wavelet-based blind watermarking of 3d models. In *International Multimedia Conference* (2004).
- [VSI00] VISWANATH N., SHIMADA K., ITOH T.: Quadrilateral meshing with anisotropy and directionality control via close packing of rectangular cells. In *9th International Meshing Roundtable* (2000).
- [WHL\*07] WANG H., HE Y., LI X., GU X., QIN H.: Polycube splines. In *ACM Solid and physical modeling* (2007).