Eurographics Symposium on Geometry Processing 2009
Marc Alexa and Michael Kazhdan
(Guest Editors)

Volume 28 (*2009*), *Number 5*

Localized Quadrilateral Coarsening

Joel Daniels II[1], Claudio T. Silva[1,2] and Elaine Cohen[1]

[1] School of Computing, University of Utah, United States
[2] Scientific Computing and Imaging Institute, University of Utah, United States

Abstract

*In this paper we introduce a coarsening algorithm for quadrilateral meshes that generates quality, quad-only connectivity during level-of-*coarsening *creation. A novel aspect of this work is development and implementation of a localized adaptation of the* polychord *collapse operator to better control and preserve important surface components. We describe a novel weighting scheme for automatic deletion selection that considers surface attributes, as well as localized queue updates that allow for improved data structures and computational performance opportunities over previous techniques. Additionally, this work supports optional and intuitive user controls for tailored simplification results.*

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

While there exist many triangle-based techniques for surface simplification, there are few results on automated algorithms for quad meshes, in part due to the unique challenges associated with quad-based geometry processing. A quad is fundamentally more complex, being a potentially non-planar and non-convex bilinear element, as compared to triangles, where both convexity and planarity are guaranteed by construction. Furthermore, unlike triangle-based methods, the structured nature of quad meshes tends to force global constraints on mesh connectivity. For this reason, existing quad-based simplification algorithms [DSSC08, SBS08] have shown that the deletion of a single quad may require the removal of a larger collection of elements to preserve an all-quad mesh or to maintain mesh structure.

Unfortunately globalized coarsening operations have adverse effects on element sizing and preservation of surface attributes. An important challenge of mesh simplification is the controlled degradation of elements to achieve an adaptive sampling that respects important surface features. In the conventional sense, this translates to the preservation of important surface geometry [GH97], but also extends to appearance attributes [Hop99], i.e. color and material properties. In contrast to triangle mesh approaches, quad mesh algorithms may also have the added complexity of achieving element alignment to orthogonal vector fields defined over the surface, i.e. principal curvature directions. To our knowledge, no previous quad-only scheme considers these properties.

Our work simultaneously addresses the challenging problem of developing attribute-aware meshes while producing high quality quad-only connectivity at all levels-of-detail, illustrated in Fig. 1. In this image, the *wooden fish* model is coarsened based on two different associated importance map attributes, where darker elements correspond to lower weights. The mesh hierarchy contains well-shaped elements, with few extraordinary vertices (non-valence 4), and sample density and alignment based on the input attribute data.

Contributions. In this paper, we describe a novel attribute-based quad coarsening algorithm, termed *qCoarsen*. The contributions of this research are three-fold: (1) A novel localized collapse template, based on the dual *polychord* operator, limits the propagation of element deletions; (2) A novel weighting scheme facilitates algorithmic decisions and improves user controls while assessing new quality metrics including element alignment to surface attributes; (3) Localized queue updates allow improved data structures and augment computational performance. Through the use of local deletion operators and a novel weighting metric, *qCoarsen* generates mesh hierarchies that exhibit controlled element size gradation while maintaining quad-only connectivity.

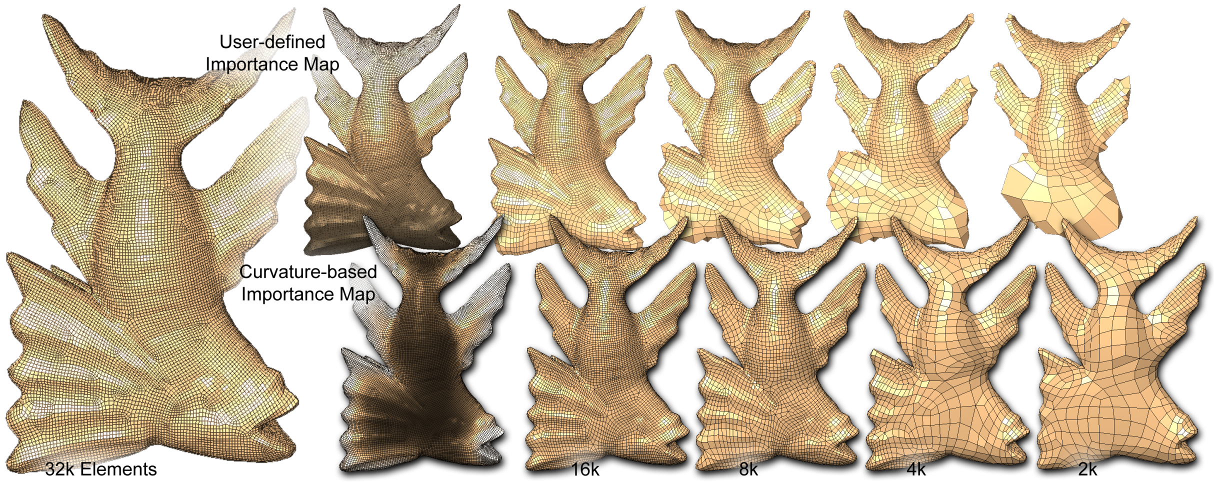

Figure 1: *Our localized quad coarsening (*qCoarsen*) algorithm generates quad-only mesh hierarchies that are sensitive to attribute data, i.e. elements scaled based on user-defined (top) or automatically computed curvature-based (bottom) attributes.*

2. Related Work

The recent interest in quad-based surface representations has resulted in the development of many construction and processing techniques. While triangle-based geometric processing algorithms have been thoroughly studied, fewer techniques exist that directly address quad meshes. In the following section, we review quad remeshing and related geometric processing algorithms.

Reconstruction techniques constitute the majority of quad-based geometric processing research. The most straightforward methods, conversion-based approaches, apply splitting [CC78] and merging [LKH08] operations to polygonal elements to construct quads. Advancing front algorithms [OSCS99] propagate a frontier curve over the model defining a faceted surface in its wake. Quad-based resampling schemes [VSI00] use rectangularly packed repulsion forces to distribute points over the model, producing similar vertex sampling as advancing fronts without the degeneracies related to front collisions. Surface grafting [JBSM99] robustly constructs quad meshes by embedding an orientable surface within a hexahedral grid. Numerical integration approaches trace iso-curves through orthogonal vector fields [KNP07, DKG05], that, when guided by principal curvature directions [MK04, ACSD*03], achieve feature alignment. Divide-and-conquer techniques also leverage path tracing over the input surface to segment the surface then individually remesh each region [CSAD04, NGH04, DBG*06, TACSD06, HZM*08]. Surface parameterization [SPR06], mapping a model to a well-known domain, facilitates the construction of structured quad meshes by appropriately sampling the parametric domain.

Quad remeshing algorithms can build level-of-detail hierarchies, mimicking the effects of mesh simplification, by tuning parameters to control element sizes. However, these approaches are not developed for mesh simplification and their application to this end is not always straightforward. Further computation is necessary to obtain the continuous transition between the hierarchy levels as exhibited by progressive meshes [Hop96]. It is unclear that the techniques are robust to handle construction of very coarse models.

Mesh Simplification. Simplification methods execute deletion operations to reduce the number of elements until breaching a prescribed error threshold. Triangle-based simplifications generate an automated prioritization method using a quadric error metric (QEM) [GH97] that defines a measurement tool to compare and minimize the collapse affects on the geometric structure. The QEM matrix encodes the planar equations of the triangles to store geometric data per vertex or edge. An extension of QEM simplification considers additional metrics, vertex valences, in order to produce quality element through simplification [SBM05].

Quad mesh coarsening is achieved as a byproduct of mesh improvement schemes, leveraging localized remeshing templates [SC97, Kin97]. Meanwhile, other improvement algorithms explore the extension of concepts developed for hex meshing, mainly the use of the dual representation [JBSM99]. These quad-based schemes extend hex-sheet extractions as *ring* deletions to augment vertex valences on grafted surfaces [BPJH02], and restructuring techniques to localize the ring deletions [SBS08] for mesh coarsening.

The research presented in this paper is most related to the quad mesh simplification algorithm (*QMS*) of Daniels et al. [DSSC08]. *QMS* is a fully automatic technique for quad mesh simplification using quality metrics that measure the impact on vertex valences, geometric loss and mesh area associated with the collapse. A generalization of the *ring* collapse, the *polychord* collapse, further discussed in Sec. 3 enables high quality simplification results. In this paper,

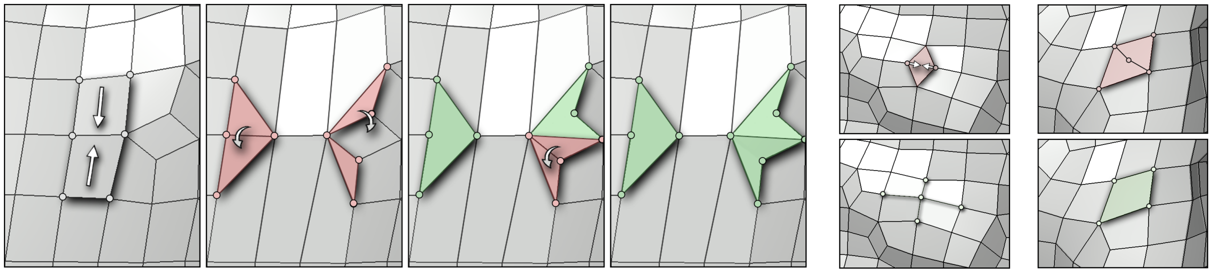

Figure 2: *Localized deletions: the quad-edge merge (*qeMerge*), quad-vertex merge (*qvMerge*) and doublet collapse. To maintain a quad-only neighborhood, the* qeMerge *includes a remeshing phase to remove triangles, illustrated above.*

qCoarsen incorporates multiple improvements over *QMS*, considering new metrics to achieve attribute sensitivity, better geometric fidelity and improved execution (Sec. 5).

3. Deletion Templates

The dual representation is a derived structure of quad meshes [BBS02], on which important operations can be performed [DSSC08]. The dual representation is defined to have the following components: the dual of a quad element is its *centroid*; the dual of a quad edge is the *chord* that connects the centroids of neighboring quads; the dual of a vertex is the *polygon* formed by connecting the centroids, in a cyclic order, of the neighboring quads.

The *polychord* is a higher-order structure, a polyline whose adjacent segments are chords that meet at a common centroid and are dual to opposing edges in that quad. On a closed mesh without boundaries, every polychord forms a closed loop. That is, starting at a single edge on the mesh and traversing opposite edges of adjacent quads, the path will always end at the starting edge. This higher-order dual structure is related to the *dual sheet* associated with hex meshing [JBSM99] for which the looping property is proved.

The dual representation is a powerful tool, useful in assessing well-behaved quad mesh surfaces. For instance, each quad of a closed 2-manifold will have exactly 4 *unique* dual chords. This property implies that every pair of adjacent quads will have at most one chord connecting their centroids, thus sharing at most one edge. This property also prevents non-manifold edges, where more than two quads share a common edge resulting in more than 4 chords per face.

As discussed earlier, many quad-based processing algorithms implement polychord collapses, simultaneously merging the vertex end points of all mesh edges to which the polychord is dual. In these works, the polychord collapse is shown to be instrumental in developing and maintaining high quality quad meshes. Unfortunately, the polychord often spans a large portion of the mesh, so that its deletion has a global effect. These structures may describe a significant portion of the model, with complex knots (see [DSSC08]) or cause the deletion of important features (Fig. 6). Instead, the following operators, (Fig. 2) are intended to localize the modification effects to avoid the complications associated with global deletions while iteratively reproducing polychord collapse configurations.

Quadrilateral Edge Merge. (*qeMerge*) This deletion operator merges two adjacent quads to their shared edge then remeshes neighboring elements to maintain the quad-only connectivity. Merging two quads to a shared edge generates an even number of triangular elements (either 0, 2 or 4). The remeshing phase (Fig. 2) is able to remove the triangles by examining the edges that emanate from the endpoints of the merged edge. Similar to the wandering edge swaps that occur in Delaunay triangulation [Wat81], the process can be described as rotating the edge shared by a triangle and quad until two triangles are adjacent and combine to form a single element. The remeshing is limited to the 1-ring neighborhood of the merged edge and is executed once for each pair of triangles, deleting 4 elements with each *qeMerge*.

This localized deletion method can reproduce the results of the polychord deletions (Fig. 3). The *qeMerge* initiates a *zipper-like* effect that locally reproduces the deletion results of two parallel polychords. In this way, the method may be used on structured models to maintain high quality connectivity. But, unlike the polychord deletion, each iteration is localized to avoid the propagation of the deletion by terminating the *zippering* effects as determined by the surface attributes, i.e. important geometry detected as high curvature.

Quadrilateral Vertex Merge. (*qvMerge*) This technique has been previously termed a quad collapse [DSSC08] and quad close [Kin97]; we use the term *quad-vertex merge* to further differentiate between the previously described quad-edge merge. This collapse method deletes a single quad by merging diagonally opposing vertices. If the quad is thought of as two virtual triangle elements connected by an edge between the merging vertices, the *qvMerge* is a generalization of the triangle edge collapse.

Doublet Removal. A doublet consists of two neighboring quads that share two consecutive edges. The vertex at which

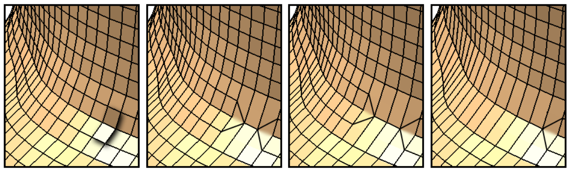

Figure 3: *The* qeMerge *followed by multiple* qvMerger*s executes a* zippering *effect that reproduces polychord collapses.*

these edges meet is valence 2 and describes a degenerate dual polygon. Doublets are removed by merging the quads into a single element.

4. Localized Coarsening

Our coarsening algorithm, *qCoarsen*, requires an input quad mesh and optional annotated feature edges, importance map, and alignment vector data. An annotated feature is a linked list of connected mesh edges, who's preservation is important, i.e. sharp features, illustrated in Fig. 5. In the absence of user-defined importance and vector attributes, the principal curvature magnitudes and directions are approximated over the discrete model [ACSD*03] by virtually subdividing each quad into four triangles. In practice the best results are observed when the scalar importance values are transformed into the range [1.0,3.0]. The simplification process is guided by the attribute data, preserving the topology of feature edges while generating simplified models that scale element sizes based on the importance map and improve element alignment to the given vector field.

Prioritizing Operations. To improve and maintain high quality mesh connectivity while obtaining attribute-awareness, it is important to intelligently select the elements for deletion. To this end, *qCoarsen* implements a single priority queue to sort the *qeMerge* and *qvMerge* operations based on the impact of each deletion on the resulting mesh. An error value is assigned to every collapse possibility for each quad and sorted within the priority queue. When an element is deleted from the mesh the subsequent entries in the priority queue must be removed or ignored. The weighting function, E, influenced by [SBM05], evaluates the vertex valence and surface attributes,

$$E(\cdot) = (\frac{V_{before}(\cdot)}{V_{after}(\cdot)} + \alpha)^2 (D(\cdot) + \alpha)^2 (\frac{A_{before}(\cdot)}{A_{after}(\cdot)} + \alpha)^2,$$

where V measures the ratio of ideal vertices to total vertices, before (V_{before}) and after the deletion and doublet removals (V_{after}); D evaluates the average distance between the merging vertices weighted by the importance scalars; and, A measures the quad alignment to the attribute vector fields, before (A_{before}) and after (A_{after}) the deletion. The constant α (in practice 0.01) is added to each term to ensure that the error metric is greater than 0, critical in the absence of importance or vector alignment data.

The ratio of *before* versus *after* ideal valences improves the mesh connectivity. For a quad mesh, the ideal valence is 4 and non-ideal vertices are *extraordinary* as they complicate parameterization solutions and subdivision schemes. This term contributes lower error values for collapses that improve the mesh structure while penalizing those that deteriorate the connectivity.

The distance importance term D causes the element gradation illustrated in Fig. 1. For the *qeMerge* operation, D is the average distance between the midpoints of the two pairs of merging edges, measured as a percentage of the bounding box diagonal and weighted by the importance attributes for each quad. In contrast, for the *qvMerge* operation D measures the importance weighted distance percentage between the two merging vertices. In this way, D computes low errors for small elements with less importance.

The alignment term A generates low error terms for collapses that improve the element alignment to associated attribute vector fields. A_{before} measures the average angle of separation prior to the collapse, modulo 90°, between the images of dual chords of the deleting quads and their associated attribute vectors projected on the normal planes evaluated at the quads' centroids. The measured angle is scaled over the interval $[0,1]$, where 0° and 90° equal 1 and 45° equals 0. For the *qeMerge* this consists of the two quads being removed, while the *qvMerge* considers a single element. A_{after} computes the angle of separation between the edges to which the quads collapse with the attribute vectors projected onto the same normal planes.

The effects of attribute vector fields are illustrated and em-

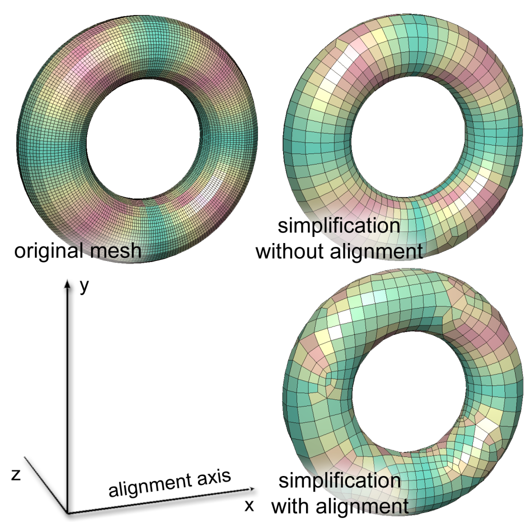

Figure 4: qCoarsen *weights attribute vector field alignment. This proof of concept example exaggerates the effects by forcing alignment to an unnatural vector field (the x-axis projected onto the surface).*

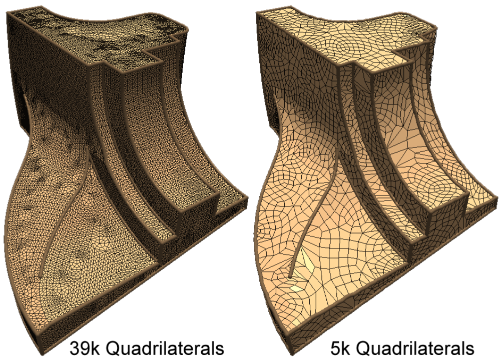

Figure 5: *Validation methods maintain the* topology *of annotated features, i.e. sharp edges, throughout simplification.*

phasized in Fig. 4 for simplification of the torus model. The elements are colored based on their alignment, computed as previously described, to a constant vector field in the x-direction. Such a vector field is not typical, but is useful in illustrating the effects of the alignment on the simplification results. In Fig. 4, *red* designates mis-aligned quads with angle of separation between the dual chords and assigned attribute vectors near 45°; and, *green* represents well-aligned elements, corresponding to angles measuring 0° or 90°. Simplification without vector field alignment is achieved by assigning null attribute vectors to the mesh.

To prevent the creation of many extraordinary vertices with high worst case valences, a conditional term is added to E. If the difference of the worst case valence count from the ideal after a collapse is greater than before, $|4 - V^w_{after}| > |4 - V^w_{before}|$, then

$$E(\cdot) = E(\cdot) + \beta \cdot |4 - V^w_{after}|,$$

where β (in practice 100) is a large constant. In this way, large error terms are awarded to collapse scenarios that greatly degrade the quality of the mesh structure.

Geometric Fidelity and Element Quality. To preserve geometric fidelity, we use an extension of QEM similar to the one used in [DSSC08] to assist the re-location of simplified vertices, reducing the error incurred by each operation. To improve element quality and attribute sensitivity during simplification, a centroidal-based smoothing procedure may be executed on the vertices belonging to the quads in the 1-ring neighborhood of each collapse. For a vertex v with neighboring quads $\{q_i\}$ and dual centroids $\{c_i\}$, surface normals $\{n_i\}$ and importance attributes $\{a_i\}$, the new point $\tilde{v}$ is evaluated as the weighted average of the centroids, $\tilde{v} = (\frac{\sum a_i c_i}{\sum a_i})$. This point is projected to the tangent plane, $\sum n_i$, defined at v to better preserve the surface geometry. Alternatively, a post-simplification global smoothing method reduces the queue updates during simplification to the 1-ring neighborhood of each collapse that improves computational performance.

|Q|	Time (sec.)	S.Jacobian (Median,Worst)	Vertex Valence (Ideal,|Ex|,Worst)	Error (10^{-3})
Wooden Fish (Fig. 1)				
33k	n/a	(0.99, 0.44)	(97%, 876, 6)	n/a
16k	20	(0.98, 0.3)	(96%, 673, 6)	$13.9d_B$
8k	30	(0.95, 0.14)	(94%, 509, 6)	$11.9d_B$
4k	35	(0.92, −0.17)	(91%, 363, 6)	$20.5d_B$
2k	37	(0.91, 0.18)	(89%, 220, 6)	$23.0d_B$
Egea (Fig. 7)				
27k	n/a	(0.63, 0.02)	(51%, 13.1k, 11)	n/a
3k	21	(0.91, 0.42)	(80%, 592, 6)	$17d_B$
12k	n/a	(0.97, 0.52)	(95%, 592, 6)	$11d_B$
Stanford Bunny (Fig. 9)				
22k	n/a	(0.92, 0.0)	(56%, 9.6k, 6)	n/a
5k	22	(0.96, 0.52)	(93%, 337, 6)	$11d_B$
1k	27	(0.93, 0.44)	(88%, 120, 6)	$31d_B$
Fertility Model (Fig. 9)				
22.5k	n/a	(0.65, 0.09)	(51%, 11k, 12)	n/a
5k	17	(0.87, 0.22)	(72%, 1396, 7)	$8d_B$
2k	21	(0.84, 0.11)	(81%, 366, 6)	$25d_B$

Table 1: *Simplification results (time, Scaled Jacobian data, mesh structure (ideal/extraordinary vertices, and worst case valence), and error) and the subdivided Egea* 12*k remesh.*

Topology Preservation. During simplification, *qCoarsen* maintains well-behaved surfaces by disallowing collapses that generate non-2-manifold models. This analysis explores the dual mesh of intermediate constructions created by each deletion, *qeMerge* or *qvMerge* followed by doublet removals. If every quad contains exactly 4 dual chords then the intermediate neighborhood is committed to the mesh, removing a number of elements.

Feature Edges. For some models and applications it may be important to maintain the topology of a number of important feature edges, i.e. sharp edges, throughout the mesh hierarchy, as illustrated in Fig. 5. Three preservation cases are used to validate an intermediate representation of the collapse neighborhood prior to committing the results. First, two merging feature points must be consecutive vertices of a common feature curve to avoid *feature pinching* which forms new feature corners. Second, the collapse must not split a feature curve into two distinct curves. Lastly, the collapse can not completely remove a feature curve.

5. Results & Discussion

The *qCoarsen* algorithm as described in Sec. 4 was implemented in C++ and running times reported in Table 1 were performed on a 2.2 GHz Dual Core AMD Opteron Processor 275 with 4GB memory. In these results, the vertex smoothing is performed once at the end of the simplification routine, as opposed to following each deletion step, typically resulting in a 4x speedup. The algorithmic cost of our *qCoarsen* implementation is $O(mk \log n - m)$ for m deletions, k updates per deletion where queue updates are $O(\log n)$ for n

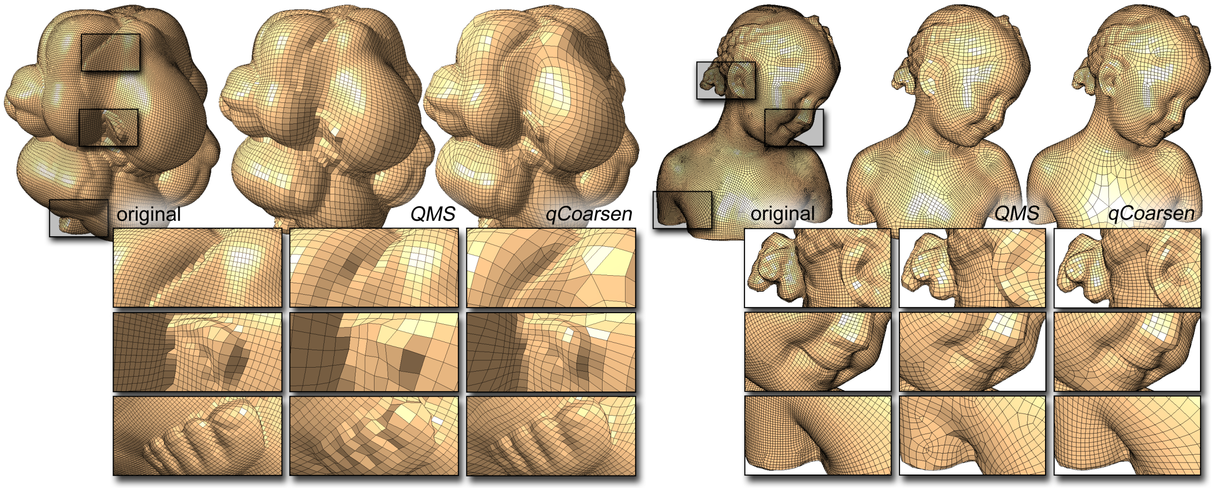

Figure 6: *Comparison of* QMS*, a quad simplification algorithm, and* qCoarsen *results at the same number of elements.* QMS *maintains high structure on semi-regular models (Pensatore), while* qCoarsen *trades additional extraordinary vertices for lower approximation errors. On other models (Bimba),* qCoarsen *generates better simplification results in every aspect (Table 2).*

|Quads|	Time (sec.)	Vertex Valence (Ideal,	|Ex|,	Worst)	Error (10^{-3})
Pensatore (Fig. 6)					
43650	n/a	99%	8	3	n/a
11k (*QMS*)	46	99%	8	3	$8d_B$
11k (*qCoarsen*)	38	99%	123	6	$1.3d_B$
Bimba (Fig. 6)					
62842	n/a	99%	726	6	n/a
15.5k (*QMS*)	479	94%	898	6	$4.5d_B$
15.5k (*qCoarsen*)	56	98%	432	6	$3.8d_B$
Wooden Fish (Fig. 1)					
32.4k	n/a	97%	876	6	n/a
2k (*QMS*)	120	84%	314	6	$56.0d_B$
2k (*qCoarsen*)	37	89%	220	6	$23.0d_B$
Bumpy Torus (Fig. 9)					
95256	n/a	50%	47.6k	13	n/a
23.5k (*QMS*)	878	66%	8006	10	$14.8d_B$
23.5k (*qCoarsen*)	69	66%	7888	9	$9.7d_B$

Table 2: *Quantitative comparison of* QMS *and* qCoarsen.

quads. Because each iteration of the deletion triggers a 2-ring neighborhood update (1-ring if a post-process smoothing is preferred), a pathological mesh may be organized such that this region includes the entire model, thus $k = n$. However, in practice the update neighborhood is much smaller, $k < 40$, where the cost is $O(m \log n)$.

The code has been rigorously tested by constructing mesh hierarchies, for various models, further analyzed in Table 1. The table quantifies the quality of the meshes, measuring statistics of the quads' Scaled Jacobians; the structure, documenting the percentage of ideal vertices, the number of extraordinary vertices and the worst case valence information; and lastly, the approximation error measuring the Hausdorff distance between the simplified mesh and its original in terms of the bounding box diagonal d_B. The *qCoarsen* algorithm generates well-behaved, homeomorphic surfaces despite the quality of the input mesh; most notably, significantly improving the mesh structure of unstructured models, while maintaining close approximations to the original.

Simplification Comparison. A comparison of the *qCoarsen* results to a previous automated quadrilateral mesh simplification algorithm, *QMS* [DSSC08], illustrates the advantages of our localized approach. In *QMS*, the use of the global polychord deletion operator allows the algorithm to exploit highly structured dual representations of some quad meshes. A subset of semi-regular quad meshes, including polycube-based remeshes [WHL*07], dual contour surfaces [BPJH02], swept and rotated geometries, and some spline- and morse-based models [DBG*06], exhibit such structure. For these meshes, *QMS* generates highly structured models through simplification with quick computations because of the vast number of quads being removed during each deletion, exemplified by the simplification of the Pensatore model in Fig. 6 and Table 2. At the expense of new extraordinary vertices, *qCoarsen* generates more accurate simplifications with adaptive sampling characteristics. Furthermore, due to the structured nature of the dual representation, *QMS* is extremely efficient, removing many quads with each pass of the polychord deletion; however, as illustrated by the timings for the other examples, this behavior is atypical.

For many quad models, all irregular and many semi-regular meshes, where the dual representation does not demonstrate the necessary structure that lends itself to polychord deletions, the advantages of *qCoarsen* are more pronounced. For example, the Bimba model, illustrated in Fig. 6, is a result of state-of-the-art quad remeshing that is

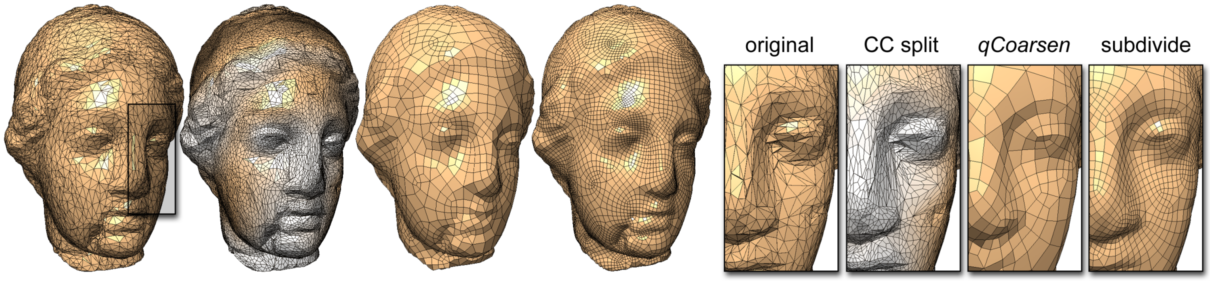

Figure 7: *Simplification-based remeshing applies* qCoarsen *to a Catmull-Clark split mesh, yielding attribute-aware, quad-only base domains that may be subdivided and projected to the original surface.*

dominated by ideal vertices (99%). However, the extraordinary vertices do not describe a coarse cube-like decomposition of the model and the dual polychords describe complex knots. Under such circumstances, *qCoarsen* outperforms *QMS* in every way when compared at the same element counts (Table 2) with faster computations, fewer extraordinary vertices, and lower approximation error (in terms of the bounding volume diagonals d_B) because adaptive sampling better describes complex geometric details.

Attribute-aware Quad Remeshing. The simplification-based reconstruction pipeline converts the polygonal mesh, splitting each element based on Catmull-Clark subdivision, into a quad-only model, then simplifies to the desired element count [DSSC08]. This technique is based on the observation that a single iteration of Catmull-Clark subdivision yields quad-only meshes despite the polygonal types of the original model. We extend this pipeline by computing curvature-based attributes over the input model, illustrated by the color scale in Fig. 7, and apply *qCoarsen* reducing to a quarter of the original element count. The simplified mesh is subdivided and projected to the original surface using a bounding volume hierarchy to improve computational performance. Because this pipeline is connectivity-based, and relies on localized deletion operations, we argue that it is both robust and simple to implement, Fig. 8.

Limitations. *qCoarsen* strives for automated attribute-aware simplification results, i.e. scaled element sizes and vector field alignment. The smoothing procedure does not consider alignment properties; consequently, when applied to remeshing, the results do not rival the alignment of global numerical integration-based methods [ACSD*03, HZM*08]. Future simplification-based remeshing may explore a hybrid scheme, connectivity-based and path tracing operators, as well as global considerations to align extraordinary vertices.

6. Conclusion

Our *qCoarsen* algorithm describes a fully automated technique for quad-based simplification that simultaneously generates well structured quad-only meshes, dominated by ideal, 4, valence vertices, while exhibiting augmented attribute-awareness via controlled element sizing and alignment to geometric features. User interaction is optional and straightforward, including an importance map and a vector field defined over the surface, to enable tailored algorithmic behavior. The localized deletion operators, while improving the attribute-awareness of the simplification results, further motivate the robustness and simplicity of connectivity-based approach to quad-only mesh processing.

Acknowledgments. We would like to thank Matt Berger, Tiago Etiene, Carlos Scheidegger and 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

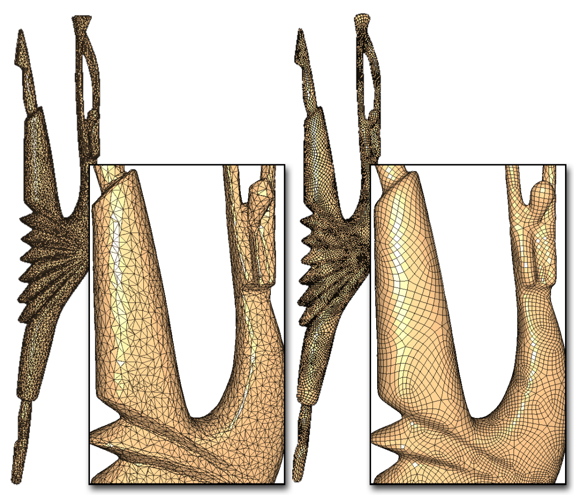

Figure 8: *The simplification-based remeshing is robust, handling arbitrary polygonal types and genus.*

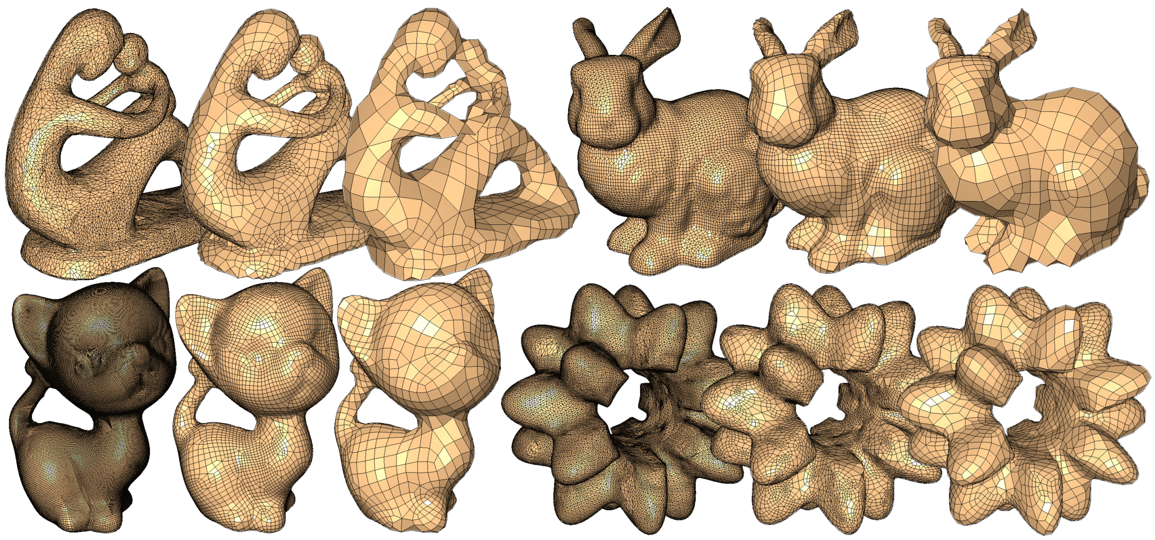

Figure 9: *Simplification results for various models of arbitrary genus at different levels-of-detail.*

ACM SIGGRAPH (2003).

[BBS02] BORDEN M., BENZLEY S., SHEPHERD J.: Hexahedral sheet extraction. In *11th International Meshing Roundtable* (September 2002).

[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).

[CC78] CATMULL E., CLARK J.: Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer Aided Design 10*, 6 (1978).

[CSAD04] COHEN-STEINER D., ALLIEZ P., DESBRUN M.: Variational shape approximation. In *ACM SIGGRAPH* (2004).

[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).

[Hop96] HOPPE H.: Progressive meshes. In *ACM SIGGRAPH* (1996).

[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., KOBBELT L., BAO H.: Spectral quadrangulation with orientation and alignment control. In *ACM SIGGRAPH Asia* (2008).

[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).

[KNP07] KALBERER F., NIESER M., POLTHIER K.: Quadcover: Surface parameterization using branched coverings. *Computer Graphics Forum 26*, 3 (2007).

[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).

[MK04] MARINOV M., KOBBELT L.: Direct anisotropic quad-dominant remeshing. In *Pacific Graphics* (October 2004).

[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).

[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).

[SPR06] SHEFFER A., PRAUN E., ROSE K.: Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis. 2*, 2 (2006).

[TACSD06] TONG Y., ALLIEZ P., COHEN-STEINER D., DESBRUN M.: Designing quadrangulations with discrete harmonic forms. In *Symposium on Geometry Processing* (2006).

[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).

[Wat81] WATSON D.: Computing the n-dimensional delaunay tessellation with application to voronoi polytopes. *The Computer Journal 24* (1981).

[WHL*07] WANG H., HE Y., LI X., GU X., QIN H.: Polycube splines. In *ACM Solid and Physical Modeling* (2007).