Template-based Quadrilateral Meshing

J. Daniels II[1], M. Lizier[2], M. Siqueira[3], C. T. Silva[1], and L. G. Nonato[4]
[1] SCI Institute, University of Utah, United States of America
[2] Departamento de Computação, Universidade Federal de São Carlos, Brazil
[3] Departamento de Informática e Matemática Aplicada, Universidade Federal do Rio Grande do Norte, Brazil
[4] Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Brazil

Abstract

Generating quadrilateral meshes is a highly non-trivial task, as design decisions are frequently driven by specific application demands. Automatic techniques can optimize objective quality metrics, such as mesh regularity, orthogonality, alignment and adaptivity; however, they can not make subjective design decisions. There are a few quad meshing approaches that offer some mechanisms to include the user in the mesh generation process; however, these techniques either require a large amount of user interaction or do not provide necessary or easy to use inputs. Here, we propose a template-based approach for generating quad-only meshes from triangle surfaces. Our approach offers a flexible mechanism to allow external input, through the definition of alignment features that are respected during the mesh generation process. While allowing user inputs to support subjective design decisions, our approach also takes into account objective quality metrics to produce semi-regular, quad-only meshes that align well to desired surface features.

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

1. Introduction

Generating quad meshes is a central problem in current geometry processing research, as many important applications, including texture and spline-based surface modeling, greatly benefit from a quad structure. Animation artists, in general, prefer quad over triangle meshes. This is partially attributed to the ability of quad elements to naturally align to principal curvature directions and feature curves that, for example, facilitates the modeling of character limbs. Further, joints, articulations and skin bends can be modeled with quad edge polylines ending in extraordinary vertices (i.e., vertices of valence other than 4), because these polyline structures behave like a hinge to improve deformations.

Although quad meshes are highly desired for the aforementioned applications, designing fully automatic techniques to produce suitable quad meshes is still a difficult task. This is partially due to the fact that fundamental quality criteria, i.e., mesh regularity (dominated by valence 4 vertices), orthogonality (rectangular quad elements), alignment (respecting principal curvature and surface features), and adaptivity (relating quad sizes to local curvature variations and anisotropy), can be in conflict with each other. Furthermore, application specific criteria, for which the mesh is targeted, may require design decisions that cannot be automatically anticipated.

On the one hand, there are well-known automatic techniques to describe the surface features of a model, and some of them have been incorporated within fully automatic quad mesh generation approaches. On the other hand, subjective design decisions require user inputs. There are a few quad meshing approaches that offer some mechanisms to include the user in the mesh generation process. However, these approaches may not ensure the desired mesh alignment with surface features; may not provide necessary or easy to use inputs to control subjective design decisions; or may require a large amount of user interaction.

This paper describes a new approach for generating all-quad meshes from triangle surfaces. Our approach offers a flexible mechanism to allow external input, through the definition of surface features that are

respected during the mesh generation process. Input feature information can be defined by the user, automatic algorithms, or a combination of both, varying the level of user involvement. While considering user inputs to support subjectivity, our approach produces semi-regular, quad-only meshes that align to surface features, conforming to the fundamental quality criteria.

Contributions. The proposed approach consists of three main stages: base triangulation construction, template-based meshing, and mesh quality improvement (Fig. 1). The base triangulation construction relies on a novel scheme that derives a triangulation from harmonic functions defined on the input model. In this way, our algorithm is oblivious to the 3-dimensional (3D) space, processing on the surface. Template meshes, defined on a triangle, are mapped to the elements of the base triangulation. A graph-based matching optimization ensures that the templates are mapped in a coherent manner, respecting the input features, constructing all-quad connectivity, and favoring orthogonality and planarity of template matching. Finally, a vertex-based optimization scheme is applied to the quad mesh to improve the quality of the final elements. We summarize the contributions of our work:

1. A novel algorithm for base triangulation construction,
2. Template definitions that generate all-quad meshes aligned to input features,
3. Novel application of a graph matching algorithm to arrange templates on the base triangulation,
4. A novel metric to optimize orthogonality and planarity during template arrangement.

2. Related Work

This paper describes a new approach for remeshing the triangle surface of a given input model with quadrilateral connectivity. In the past few years, several methods have been developed with the same goal in mind. In what follows, we review those methods, emphasizing the shortcomings that are addressed by our approach. For a comprehensive review on quad meshing techniques, we refer the reader to [47, 2, 21].

Quadrilateral Meshing. Driven by the successful philosophies in triangle-based meshing, many state-of-the-art techniques seek to construct high quality quadrilateral meshes without the need for user intervention. For instance, rectangular bundled repulsion potentials [52], Lp centroidal Voronoi tessellations (Lp-CVT) [31], and vertex smoothing [29] automatically distribute points in a quad-packing over the model. Anisotropic repulsion forces and distance computations allow the alignment of the final elements to an underlying vector data. Similarly guided by vector alignment, numerical integration of certain orthogonal vector fields defined over the surface, i.e. principal curvature directions [1, 34], yield quad elements aligned to geometric features. However, each of these techniques generate many extraordinary vertices and non-quad elements.

Triangle pairing schemes using greedy algorithms [51, 29, 49] are robust techniques to construct quad-dominant meshes. Such methods have been shown to build well aligned mesh edges, but may not produce all-quad models, nor align extraordinary vertices. Greedy pairing schemes do not guarantee to provide an optimal matching. Similarly, an unweighted graph matching algorithm [41] coupled with Steiner vertex insertions generates quad-only meshes; however, alignment becomes problematic. A template-based meshing approach for building quad meshes from imaging data reconstructs image features [32]. However, this technique is constrained to planar regions and uses a heuristic for pairing base triangles that does not generate mesh alignment.

Parametrization methods generate quad meshes that are dominated by idealized vertex neighborhoods (valence 4). For instance, clustering of cone singularity candidate points and conformal parametrization [4], naturally locate extraordinary vertices. Further, fitting parametrization gradients to the underlying vector field, using periodic coordinates [42], branch coverings [24], and a mixed integer solver [7], improves element alignment of the final mesh. However, aligning iso-values of the parametrization to surface features, and between extraordinary vertices is a challenging task, demonstrated in Sec. 7.1.

Semi-automatic, spectral quadrangulation [14] relies on user selected eigenfunctions to drive the number and placement of extraordinary vertices. Extension of this work provides additional controls over alignment and importance sampling [22].

While such methods produce high-quality quad meshes, eigenfunction browsing, parameter tuning, feature alignment and anisotropic sampling are challenging tasks. Similarly using the Morse-Smale complex to build the final quads, two sizing fields allow anisotropic sampling [57]; however, these models do not exhibit the same structural regularity of the spectral-based techniques.

T-spline modeling [46, 45], recently T-meshes [35], support T-junctions within the mesh. This added degree of freedom facilitates importance sampling to capture geometric details and feature alignment. However, sim-

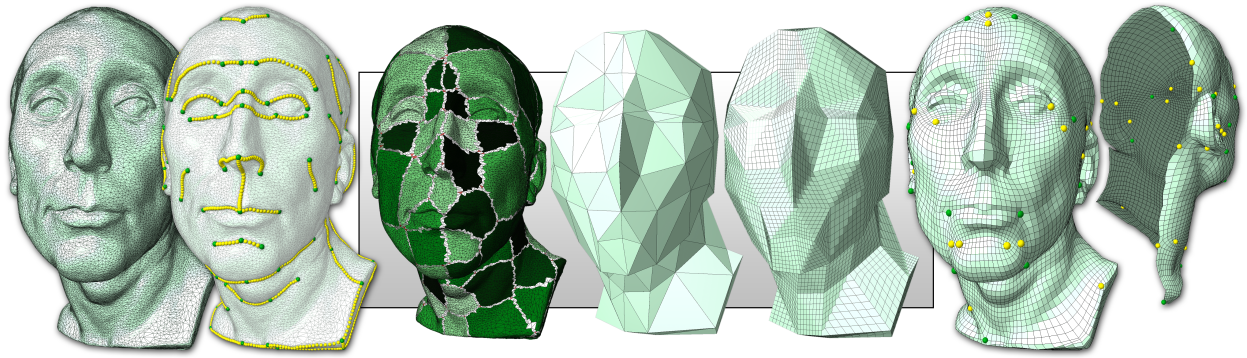


Figure 1: From an input triangle surface and feature curves (leftmost), our template-based remeshing approach defines a cell decomposition that is used to construct a base triangulation. We map template meshes to the base domain triangles, then optimize vertex locations as we map the vertices back to the original surface (rightmost).

ilar to geometry clipmaps [33] and multi-chart geometry images [8], if not properly handled, zero-area triangles can be problematic.

Vector Field Design. To this point our quad meshing discussion has assumed that the underlying vector field that guides mesh alignment is an approximation of the principal curvature directions. However, this need not be the case, as user designed fields can be seamlessly placed into many of the quad meshing methods.

Vector field design techniques provide methods to smooth potentially noisy vector fields and extraneous field topology. For instance, radial basis functions smoothly interpolate constraint vectors [40]. User interactions, providing alignment curves [54] or modifying field singularities [56], allows for subjective design decisions. User designed fields can be seamlessly placed into many of the quad meshing methods. Cross-field design [19, 36, 43] enriches the number of possible singularities. However, designing fields where numerical integration exactly traces surface features is challenging, if not impossible due to rounding errors, resulting in artifacts within corresponding quad meshes.

Base Domain Modeling. Leveraging a coarse representation of a surface model is a useful practice in geometry processing, as demonstrated by morphing applications [44, 27]. Base domains have been employed by remeshing and parametrization algorithms; constructed through refinement of hybrid meshes [18], Delaunay triangulation of distributed points [10], mesh simplification [30, 25, 38, 12, 49], clustering [11, 6], as well as user boundary painting [28, 50] and block construction [48]. The regular refinement of the base domain model generates a semi-regular mesh connectivity, beneficial to subsequent processing. Feature reconstruction is dependent on the base domain, a challenging task for some techniques. Further, user editing may be challenging or time consuming in these approaches.

In this work, we propose a method to construct a base domain model that respects a set of feature points and curves without discriminating between them. Using efficient linear system solves, our method naturally handles these inputs and provides fast feedback to allow incremental editing for an iterative construction of base domain models. Our method exactly reconstructs feature curves defined on the surface, and offers flexibility through template application that allows for animation specific mesh configurations (Fig. 11). In contrast to previous efforts, we combine several key aspects: iterative editing controls, exact feature reconstruction, and semi-regular output structure.

3. Method Overview

Our quadrilateral meshing approach comprises three main stages: base triangulation construction, template-based meshing, and mesh quality improvement (Fig. 1). First, we build a base domain triangulation constrained to a few feature points and curves defined over the input triangle surface. Feature points and endpoints of feature curves become the only vertices of the base triangulation. The construction of the base triangulation from the feature points and curves relies on harmonic fields computed over the surface (Fig. 2 and 3). This computation is intrinsically 2D, carried out on the input surface and it is oblivious to the 3D space.

An all-quad mesh is generated during the second stage: template-based meshing. The key idea is to map template meshes defined on a canonical triangle to each face of the base triangulation. We design templates (Fig. 5) that can be consistently arranged over the base triangulation to produce an all-quad mesh. To ensure that we always obtain such a consistent arrangement, we

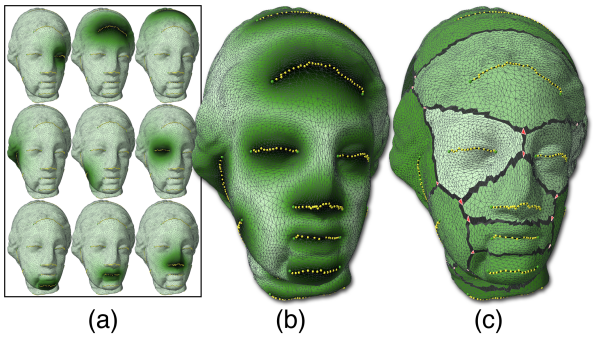


Figure 2: Harmonic functions computed for each feature curve (a) are compiled (b) to guide the segmentation of the model (c).

devised an optimization procedure based on classical results from graph theory. A distinguishing feature of our optimization procedure is that it naturally incorporates important quad mesh attributes. The set of templates can be enriched to accommodate for local changes in the mesh structure (Sec. 7).

Finally, a vertex optimization procedure improves the quality of the final all-quad mesh, moving vertices over the input surface based on the edge length ratios defined by the templates. It is important to emphasize that our approach is valuable as an iterative framework for user designed quad-only meshes. While the user identifies feature information, the method automatically manages element alignment and mesh structure. The results and design task can be augmented by automated feature identification [3, 20, 23], which alleviates user input (Fig. 10).

4. Base Triangulation Construction

The base triangulation provides the underlying space in which the templates are applied to produce the final quad mesh. The base triangulation is indeed a constrained triangulation, since each feature curve identified over the input surface is represented by an edge of the base triangulation. This section describes a method for robustly generating the base triangulation, and also explains how triangles from the input triangular mesh can be related to triangles in the base triangulation.

4.1. Feature-based Cell Decomposition

Let $\mathcal{P} = \{p_1, \ldots, p_m\}$ and $\mathcal{C} = \{c_{m+1}, \ldots, c_n\}$ be a set of *sites* and a set of *curves* on a triangle surface S, respectively, such that p_i cannot belong to any curve in $\mathcal{C}$ and $c_i \cap c_j$ is either empty or an end point of c_i and c_j, for every i, j. The algorithm for creating the base triangulation decomposes the *underlying space*, $\overline{S}$, of S (i.e., the set of all points in $\mathbb{R}^3$ spanned by the points, edges, and triangles of S) into a set of cells, $S_1, \ldots, S_n$, which defines a cell complex decomposition of $\overline{S}$, $\overline{S} = \cup S_i$ and $\text{int}(S_i) \cap \text{int}(S_j) = \emptyset$, where $\text{int}(S_i)$ is the interior of S_i. Each cell S_i corresponds to either a site in $\mathcal{P}$ or a curve in $\mathcal{C}$, and is given by the following set of points:

$$S_i = \{x \in \overline{S} \subset \mathbb{R}^3 \mid h_i(x) \geq h_j(x), \text{ for all } i \neq j\}, \quad (1)$$

where the $h_i(x)$'s are harmonic functions defined on the underlying surface, $\overline{S}$, which can be obtained by solving the Laplace equation

$$\nabla^2 h_i = 0 \quad (2)$$

with the following Dirichlet boundary conditions:

$$\begin{cases} h_i(x) = 1 & \text{if } x = p_i \text{ or } x \in c_i \\ h_i(x) = 0 & \text{if } x = p_j \text{ or } x \in c_j,\ i \neq j \end{cases}$$

Figure 2a illustrates the individual harmonic fields obtained by solving Eq. (2) for a given set of feature sites.

We discretize Eq. (2) using cotangent weights as described in [39]. Moreover, we use the CHOLMOD library [9] to solve the linear system derived from the discretization of Eq. (2). The library implements a supernodal scheme to update the Cholesky factorization [13], which efficiently supports iterative inclusion and removal of constraints via penalty schemes [55].

Identifying Cells. Once the harmonic functions h_i, for all $i = 1, \ldots, n+m$, have been obtained, the S_i's are computed as follows. Let v be a vertex of the input triangle surface S. The label k is assigned to v if $h_k(v) > h_i(v)$, for all $i \neq k$. The cell S_k contains the set of vertices of S assigned label k, as well as all edges and triangles of S spanned by these vertices. Note that each cell S_k can be decomposed as a simplicial subcomplex of S.

Each triangle t in S can be classified as belonging to a single cell S_i, being between two cells, S_i and S_j, or being in the intersection of three cells, S_i, S_j, and S_k, depending on whether the vertices of t are assigned one, two of three different labels, respectively. The cell decomposition $\cup S_i$ is fully determined by the vertex labels, which allows for a discrete and robust representation of the cell decomposition.

Unlike Dijkstra-based fast marching region growing schemes, our cell decomposition method generates similar results regardless of the underlying triangulations. Further, this harmonic-based scheme handles curves in a more natural way than geodesic-based methods. Figure 2c illustrates a cell decomposition where triangles whose vertices were assigned one (resp. two or three) distinct labels are shown in green (resp. black or red).

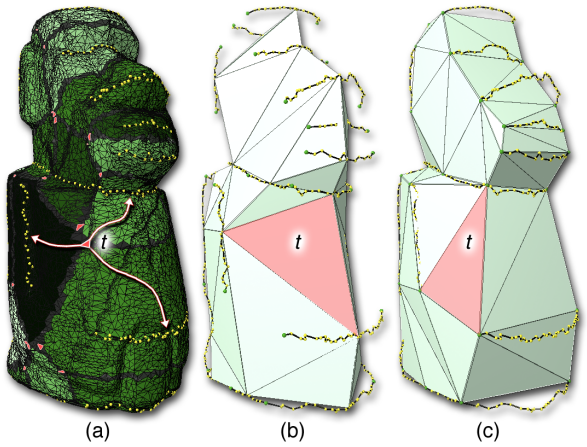


Figure 3: (a) Cell decomposition. (b) Base triangulation obtained as the dual of the cell decomposition. (c) Feature curve constraints incorporated into the base triangulation by vertex splits.

4.2. Base Triangulation

Those familiar with space partition theory will notice the resemblance between Eq. (1) and the definition of Voronoi cells. Inspired by the principles of the Delaunay triangulation [16], we define a coarse triangulation, called *base triangulation*, as the dual structure of the cell decomposition $\cup S_i$. Hereafter, we denote this base triangulation by T.

More precisely, if we view triangles of S with three distinct labels as the intersection points of the three cells (i.e., Voronoi vertices), then the triangles of T are dual to those intersection points. The vertices of T correspond to the cells defined by the features in $\mathcal{P}$ or C. In fact, if the set of feature curves C is empty, then the sites in $\mathcal{P}$ are the vertices of T. Because T is populated when three cell regions intersect at an element in S, it is necessary that such an element exists in order to construct a base domain triangulation. The union of feature sites, $\mathcal{P} \cup C$, must be greater than three.

The duality described above represents the feature curves in C as vertices of T (Fig. 3b). However, each curve in C should correspond to an edge of T. In order to build a one-to-one correspondence between curves in C and edges in T, we make use of a scheme based on vertex split (Fig. 3c). Suppose that a curve $c_i \in C$ consists of a single segment. Then, by splitting the vertex in T corresponding to c_i and moving the resulting two vertices to the locations of the endpoints of c_i, we can build a correspondence between c_i and an edge of T. The newly defined edge has the same endpoints as c_i. If curve c_i consists of multiple connected segments, then multiple vertex split operations are required.

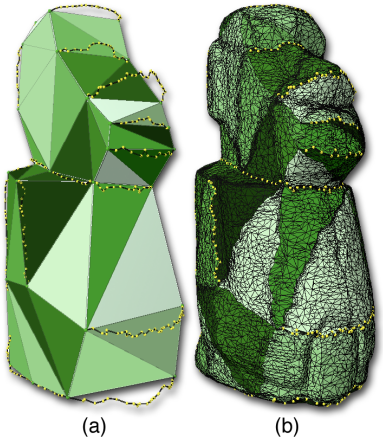


Figure 4: Input mesh is mapped to the base triangulation.

4.3. Surface Correspondence

In the second stage of our approach, during template meshing, we map points from the base triangulation T to the input surface S. To construct the correspondence between these two triangulations, we first associate each triangle t_b of T with a *geodesic triangle* t_g on $\overline{S}$. The edges of t_g are formed by three approximate discrete geodesic curves on $\overline{S}$ connecting the vertices of t_b, which are also vertices of t_g (Fig. 4). Next, we cut $\overline{S}$ along the geodesic arcs, and then modify the triangle surface S accordingly. Finally, we map *all* triangles inside t_g to t_b, establishing a many-to-one correspondence between the triangles of (the modified triangle surface) S and the ones of T. To map the triangles of t_g to t_b, we first parametrize the boundary curves of t_g in the corresponding edges of t_b by arc length, and then use the Mean Value Parametrization [15] to map interior of t_g to t_b.

5. Template-Based Meshing

Templates can be seen as textures that, when mapped to triangles in the base triangulation, T, produce an all-quad mesh. A naïve template definition, such as to split each base triangle into three quads by connecting the edge midpoints to the centroid of the triangle, introduces one extraordinary vertex per triangle, violating our goal of producing a semi-regular, all-quad mesh with only a few extraordinary vertices. In what follows, we present a new scheme to map templates to the base triangulation, T, in order to create an all-quad mesh. This scheme tries to avoid the insertion of new extraordinary vertices into the final quad mesh, while ensuring the alignment of quad elements with the input features.

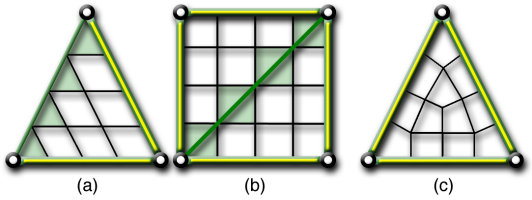


Figure 5: The regular template is designed to align with two edges in each base triangle (a). An all-quad mesh is produced by arranging the templates consistently (b). The singular template (c) resolves isolated triangles resulting from the matching scheme.

5.1. Regular and Singular Templates

Consider the template defined within the triangle in Fig. 5a, which we call the *regular template*. This template has two types of edges: *alignment* edge (i.e., the two yellow edges) and *non-alignment* edge (i.e., the dark green edge). The regular template has two important properties: 1) it is aligned with the two alignment edges of the triangle, and 2) if the templates of any two adjacent triangles match consistently, an all-quad mesh is obtained (Fig. 5b).

Property 1 can be exploited to align quads with feature curves. In fact, as each feature curve has a corresponding edge in the base triangulation, one can choose these corresponding edges to be the alignment edges of the regular template, thus naturally aligning quads to feature curves.

As we shall see in the next section, a consistent template matching (see Property 2) can always be found. However, if we wish to enforce the regular template alignment edges to be aligned with feature curves, then a consistent template matching may not exist. Edge alignment, though, is crucial for generating high quality quad meshes. So, to obtain a consistent template matching which also incorporates edge alignment, we combine the regular template with other template types (Fig. 5c).

There is at most one configuration in which a consistent template matching cannot be obtained by using *only* regular templates. In this case, all three edges of a triangle t of the base triangulation, T, are alignment edges, indicative of an isolated triangle formed during the matching process. Our approach handles this scenario by using the *singular template* (Fig. 5c), in which we introduce an additional extraordinary vertex. In the next section, we present an optimization mechanism, that relies on classical results from graph theory, to obtain an all-quad mesh using our templates.

5.2. Template Arrangement

Suppose initially that we are not interested in enforcing mesh alignment to the feature curves. In other words, we assume that the set, C, of feature curves is empty, which means that the base triangulation, T, has been generated using only the sites in $\mathcal{P}$. The problem we face is how to arrange regular templates on T to avoid the use of singular templates. Note that we can solve this problem by finding a perfect pairing of triangles of T (if one exists), then mapping the regular template to each triangle.

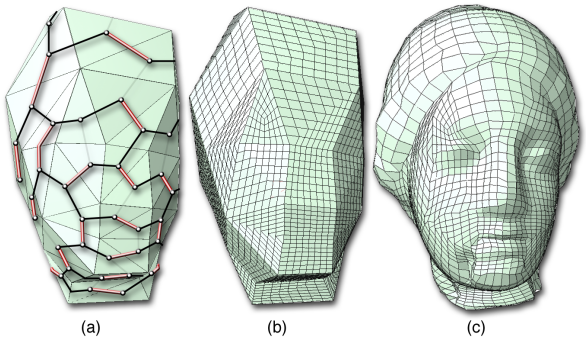


Figure 6: The graph matching template arrangement (a). Red edges define the matching of the dual graph of the base triangulation. Light gray edges have been removed from the dual graph to enforce edge alignment. Templates are consistently mapped to base triangles (b). Template vertices are mapped to the input surface based on the correspondence as discussed in Sec. 4.3 (c).

Finding a perfect pairing of triangles of T is equivalent to solving the problem of finding a perfect matching on the dual graph of T. Indeed, given a graph G, a *matching* M on G is a subset of edges of G such that no two edges of M are incident with the same node of G [26]. A matching M on G is said to be *perfect* iff all vertices of G are incident to an edge of M. In turn, the *dual graph*, G_T, of T is a graph in which every node is associated with a distinct triangle of T, and every triangle of T has a node in G_T associated with it. Furthermore, G_T has an edge connecting two of its nodes iff their associated triangles share an edge in T (Fig. 6a).

Since no two edges of a matching share a node, and since every node is incident with (exactly) one edge of a perfect matching, a perfect matching on the dual graph, G_T, of T defines a perfect pairing of the triangles of T. The theorem by Julius Peterson [37] asserts that every cubic graph (a graph in which all nodes have valence 3) with at most two bridges (edges whose removal increases the number of connected components in the graph) has a perfect matching. The dual graph, G_T, of T is clearly cubic (unless T has boundary edges), and it can be shown that G_T is bridgeless. As a result, by Petersen's theorem G_T admits a perfect matching.

Let us now assume that the set C of feature curves is not empty, and quad edges should be aligned to these curves. Therefore, the edges of G_T dual to the edges of

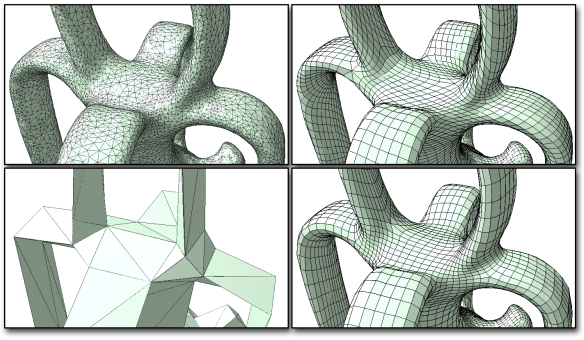


Figure 7: The twisting artifacts observed on the final quad mesh (upper right) are avoided by the orthogonality metric (lower right) used to prioritize the graph-based matching of the base domain elements.

T that correspond to feature curves should not be part of the matching. To enforce this constraint, we remove these dual edges from G_T, producing a *restricted dual graph* G'_T. Because G'_T may not be cubic nor bridgeless, Petersen's theorem no longer holds. Instead, the best we can ask for is a *maximum cardinality* matching on G'_T (maximizing the number of matched nodes) that can be computed in $O(n^{1.5})$ time [26] (n is the number of nodes).

Not every node of G'_T is guaranteed to be matched. Unmatched nodes correspond to unpaired triangles of T, to which regular templates cannot be applied. To generate an all-quad mesh, singular templates are mapped to unpaired triangles of T. Figure 6a illustrates a maximum cardinality matching on the restricted dual graph of the base triangulation of the Egea model. Matching edges are shown in red. The resulting template arrangement is presented in Fig. 6b, while Fig. 6c shows the quad mesh after the template vertices are mapped to the input triangle surface.

5.3. Orthogonality and Planarity

In addition to enforcing mesh alignment, the graph-based matching can be modified to take into account the orthogonality and planarity. We aim at improving these metrics measured on the quadrilaterals that result from the pairing of two triangles. As such, we select triangle pairings that yield quads whose angles are as close as possible to 90° and are near planar. Our approach makes similar considerations to existing triangle pairing schemes [5]; but combines the multiple factors into a single term without threshold tests.

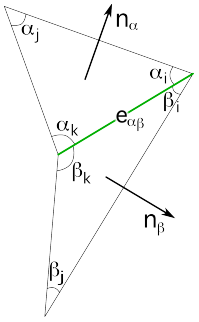
It turns out that orthogonality and planarity can be naturally incorporated in the graph matching by assigning weights to the edges of G'_T. We define a *weight function*, $w : E'_T \to \mathbb{R}$, where E'_T is the set of edges of G'_T, and then compute a *maximum weight* matching on G'_T that maximizes the sum $\sum_{e \in E'_T} w(e)$, where

$$w(e) = \frac{1}{\exp[\ \lambda \cdot \{\ 1 - (\ \mu \cdot o(e) + (1-\mu) \cdot p(e)\ )\ \}\ ]}, \tag{3}$$

$\lambda \in \mathbb{R}_+$ and $\mu \in [0,1] \subset \mathbb{R}$ are predefined constants, and $o, p : E'_T \to [0,1] \subset \mathbb{R}$ are functions that represent the degrees of orthogonality and planarity, respectively, of the quad formed by pairing up the triangles sharing the dual edge in T of edge e in G'_T. More specifically, let t_α and t_β be the two triangles of T sharing the edge $e_{\alpha\beta}$ in T, where $e_{\alpha\beta}$ is the dual in T of edge e in G'_T. The value of o at e is

$$o(e) = \frac{\sin(\alpha_j) + \sin(\beta_j) + \sin(\alpha_i + \beta_i) + \sin(\alpha_k + \beta_k)}{4}, \tag{4}$$

where α_i, α_j, and α_k (resp. β_i, β_j, and β_k) are the angles of t_α (resp. t_β), as illustrated on the right. The value of p at e is



$$p(e) = \frac{1 + \mathbf{n}_\alpha \cdot \mathbf{n}_\beta}{2}, \tag{5}$$

where $\mathbf{n}_\alpha$ and $\mathbf{n}_\beta$ are the unit normals of t_α and t_β, respectively, and $\mathbf{n}_\alpha \cdot \mathbf{n}_\beta$ is their dot product. Note that the degree of orthogonality and planarity is proportional to $w(e)$. Constant μ controls the influence of $o(e)$ and $p(e)$ over $w(e)$, while constant λ is used to scale the range of function w. In our results, $\mu = 0.5$ and $\lambda = 10$ (Sec. 7).

We used the algorithm devised by Gabow [17] to compute a maximum weight matching on G'_T, which computes the pairing in $O(n^2 \log n)$ time, where n is the number of nodes of G'_T. Figure 7 compares two quad meshes obtained using a maximum cardinality matching (upper right) and a maximum weight matching (lower right) with the weight function in Eq. (3). The mesh produced by the maximum cardinality matching presents twisted regions, which result from pairing triangles without taking into account orthogonality and planarity. The twisting effect disappears when the maximum weight matching is used, as the weights given by Eq. (3) are measures of orthogonality and planarity.

6. Vertex Optimization

The final stage of our quad meshing approach consists of a vertex optimization, which modifies template

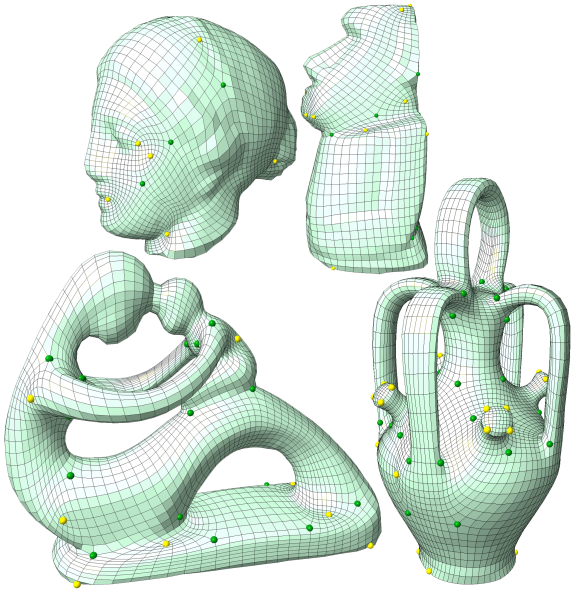
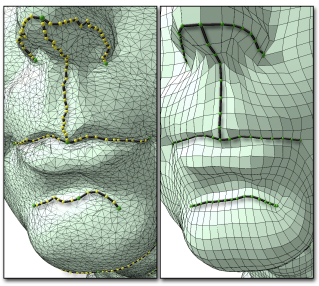


Figure 8: Quad meshes designed using our system, highlighting the extraordinary vertices in *yellow* and *green* depending on their valence count.

vertex locations in order to compensate for parametric distortions in the template mapping. To carry out this optimization, we use a standard Laplacian-based mesh smoothing approach that iteratively minimizes the difference between vertex locations and the average location of their 1-ring neighborhood vertices. The movement vectors, computed in 3D space, are projected onto the input triangle mesh, restricting the movement over the surface.

The input features, used during the construction of the base triangulation, are maintained with additional considerations. Template vertices that map to feature curves are allowed to move within a cylindrical space enveloping the curve. In this way, as illustrated to the right, we simultaneously improve the quality of quads around features, while smoothing feature noise. To maintain correctly oriented elements, movement of non-feature vertices is further restricted by limiting the motion within the boundary curve of the 1-ring neighborhood. In practice *foldovers* became problematic only in the region of constrained vertices and could be further avoided by associating a small, localized repulsion force with the feature curves.



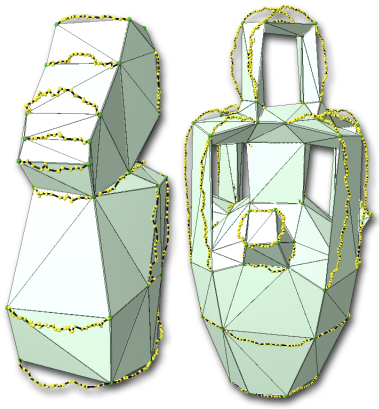


Figure 9: The complexity of the model and diligence of the user impact the time for base triangulation construction from a few seconds (left) to a several minutes (right).

7. Results and Discussion

Our quad meshing approach produces semi-regular meshes in which quad elements are well-aligned with surface features (Fig. 8). In practice, input features are symmetrically defined so as to split the model in homogeneous regions, where base triangles are more likely to be matched. As a result, unpaired triangles, which give rise to additional extraordinary vertices, tend to appear between homogeneous regions, which is highly desirable.

The quality of the quad meshes generated by our approach depends on the features from which the base triangulation is built. In some cases, features can be defined quite easily with a few curves (the Moai model, Fig. 9 left). But, some models may demand a more elaborate definition of feature curves, which can take several minutes of user interaction to be completed, (the Botijo model, Fig. 9 right). An alternative is to use automatic feature detection mechanisms (Fig. 10 top). Our approach was able to generate a reasonable quad mesh from the automatically defined features in Fig. 10. However, better alignment is obtained with additional user input, additional feature curve segments illustrated in the bottom row of Fig. 10. Since our mesh generation pipeline does not distinguish between automatically defined and user defined features, our approach turns out to be very flexible and easy to use. In addition, it can be readily combined with quad mesh design tools.

Another interesting property of our approach is the ability of locally changing the quad mesh structure. In fact, any template having two alignment edges can be mapped to triangles paired by the graph matching. To illustrate this point, we change the template mapped to the knuckles of the hand model in Fig. 11. The new tem-

plate is designed to better reproduce the configuration usually observed in quad meshes designed for character animation purposes.

7.1. Quality and Performance Analysis

Table 1 report timings and quality statistics of our approach recorded while producing the models shown throughput this paper. *Input Sizes* specifies the number of mesh vertices ($|V_m|$ is the dimension of our linear system), feature curves ($|C|$) and the total number of constrained vertices ($|V_c|$). *Algorithm Timings* times three significant stages of our approach: (1) the linear system solve for a single feature curve (*Laplace*), (2) the mapping of the input triangles to the base domain (*Mapping*), and (3) the projection of the template vertices to the input surface mesh using the mean value parametrization (*MVP*). The base triangulation (mapping and vertex splitting) and the graph matching stages are not reported in these timings because they occur near instantaneously. The output statistics include the number of triangles in the base domain models ($|T|$), the number of vertices in the final mesh ($|Quad\ V|$) and extraordinary vertices ($|ExV|$), as well as the average angle deviation from the ideal 90° ($Avg\ (|90° - < |)$).

Loading all feature curves from file at input, the base triangulation requires 42s in the worst case (the Botijo model). However, when adding feature curves incrementally within an interactive construction or editing framework, the recomputation of the linear systems can be localized a subset of the feature curves. In practice, only 5 or 6 curves are affected by curve insertion or deletion. As such, the update times of the base triangulation are reduced to below 3s, providing quicker visual feedback. This feature allows for an iterative construction process in contrast with other base triangulation and state-of-the-art quadrilateral meshing methods.

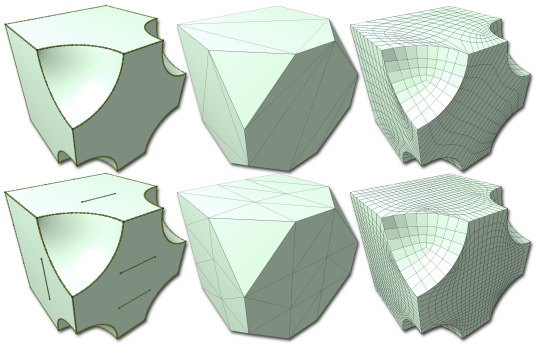


Figure 10: Automatic feature detection can be used to generate initial results (top), but may benefit from additional user interaction (bottom).

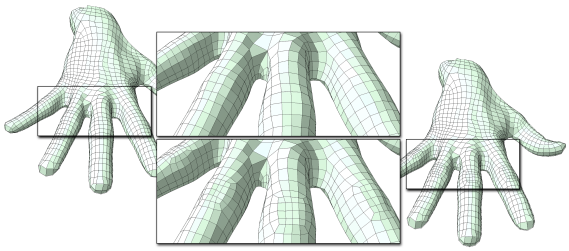


Figure 11: Interaction with the applied mesh templates changes the regular sampling of the hand knuckles (left and top) to mesh configurations observed in animation-based models (right and bottom).

A visual comparison of the Rocker Arm model (Fig. 12) illustrates the differences between recent meshing algorithms [7, 43] and our approach. On a qualitative level, we produce a similar number of extraordinary vertices (38) as [7] (36) and [43](54). Additionally, the histograms of the mesh angles demonstrate a quality on par with other approaches, despite vertex optimization not being a focus of this work. On the Rocker Arm models, the edges emanating from the extraordinary vertices are highlighted, tracing across regular vertices until reaching another extraordinary vertex. These traced edges can be used to describe boundary components of a coarse, quad-only decomposition of the model.

In this comparison, we direct the reader's attention towards two fundamental differences between our approach and existing methods. First, our meshes exhibit a regular structure with aligned extraordinary vertices, characterized by decomposing the model into a coarse number of large quadrilateral patches. Second, we align the quadrilateral elements to feature curves, exactly reconstructing sharp edges as highlighted (the misaligned elements are shown in red) in the zoom insets. These structural components are desirable for subsequent geometry processing, i.e., texture mapping and smooth surface fitting [53], and avoid geometric artifacts and feature smoothing during mesh subdivision. These structural attributes occur naturally in our approach, yet are challenging tasks for approaches that rely on numerical integration of tensor data [7, 43]. While it may be possible to modify the parametrization schemes of such approaches to ensure feature and extraordinary vertex alignment, to the best of our knowledge, this remains a challenging and unsolved task.

The structural regularity of our meshes are further demonstrated in the comparison of the Botijo model with a more recent meshing result [57] (Fig. 13). We produce a similar number of extraordinary vertices (73) versus [57] (78), which is driven by the number of identified feature curves. This comparison again de-

Table 1: Timings and quality statistics for models used throughout this paper.

Model	Input |V|	Constraints |C|	Constraints |V|	Algorithm Timings (Laplace / Mapping / MVP)	Base Domain |T|	Output Vertices (|V| / |Ex|)	Angle Statistics Avg(|90°− < |)
Botijo (Figs. 7,13)	20K	78	13K	0.53s / 7.5s / 16.3s	334	11K / 73	12°
Egea (Figs. 2,6,8)	24.9K	25	656	0.36s / 4.4s / 35.1s	96	3.2k / 22	11°
Fertility (Fig. 8)	20K	54	845	0.33s / 2.9s / 22.2s	232	5K / 43	18°
Hand (Fig. 11)	20K	68	866	0.25s / 7.1s / 13.3s	270	2.2k / 112	9°
Moai (Figs. 3,4,8,9)	10.5K	30	545	0.09s / 1.3s / 9.6s	80	2.7K / 34	12°
Nicolo (Fig. 1)	25K	73	1345	0.49s / 5.2s / 19.6s	159	5.3K / 41	13°

composes the models, by tracing between extraordinary vertices, to demonstrate the structural regularity of our model. Further, the zoom inset of the models' base demonstrates the alignment of the boundaries edges to feature curves. Additionally, the handle inset demonstrates the ability to create anisotropic elements, which, due to the regular template design, is linked to the length of the identified feature curves and the space between them.

Limitation. Guaranteeing topological equivalence between the input triangle surface and the base triangulation is one of the main limitations of our approach. This is important for the parameterization of the original model to the base domain in order to map the quad elements of the applied templates to the final surface. Despite not guaranteeing topological equivalence through the construction process, we did not face any difficulties in producing base triangulations homeomorphic to the input model. However, this could become an issue when handling complex high genus models. Adaptation of existing sampling methods, in particular farthest point insertion [10], can be implemented to automatically ensure topologically coherent base domains.

8. Conclusion

This paper describes a new approach for generating all-quad meshes from triangle surfaces. This approach uses input feature points and curves to steer the quad mesh construction, allowing for a simple mechanism to control quad mesh alignment as well as extraordinary vertex placement. The main contributions of the approach are a harmonic-based algorithm for partitioning a triangle surface mesh; a set of triangle-based, quad-dominant templates for generating quad-only meshes; the application of a graph-based algorithm to compute a consistent template arrangement on a coarse triangle mesh; and a metric for measuring orthogonality and planarity of quad elements formed by triangle pairing.

The use of harmonic functions to build cell decompositions from triangle surfaces describes a simple scheme to generate coarse triangulations. This can be very useful to other application domains, such as subdivision surfaces and surface-based remeshing. The graph-based template arrangement mechanism produces well structured and aligned quad meshes. Distinct template patterns provided the capability of generating irregular quadrilateral configurations, which are similar to those observed in animation-based meshes, a characteristic rarely found in quad mesh schemes previously described in the literature. We believe that the simplicity of the user interactions from which our method is able to generate quality meshes, presents an attractive approach towards user involvement in quad-only mesh generation.

Future Work. We believe that templates can also be arranged on the base triangulation by a graph coloring rather than a graph matching algorithm. A coloring scheme would allow the use of a rich library of templates, increasing the flexibility and ability of our approach to deal with irregular quad configurations. Additionally, we are interested in extending our harmonic-based scheme to building coarse triangulations that are homeomorphic to the input triangle surface. Beyond homeomorphism, building base models for the purpose of subdivision surfaces is an interesting (and useful) research project.

The vertex optimization scheme, a black box component of our algorithm, should be improved in order to ensure a better distribution of quad vertices. For instance, using the movement vectors as computed by Lp-CVT [31] may obtain better quality elements while considering principal curvature directions. However, to maintain the quality mesh structure it would be necessary to fix the mesh connectivity; and thus, limit vertex movements in cases to avoid inverted and non-convex elements.

References

[1] Alliez, P., Cohen-Steiner, D., Devillers, O., Lévy, B., Desbrun, M., 2003. Anisotropic polygonal remeshing. ACM Trans. Graph. 22 (3), 485–493.

[2] Alliez, P., Ucelli, G., Gotsman, C., Attene, M., 2008. Recent advances in remeshing of surfaces. In: Floriani, L. D., Spagn-

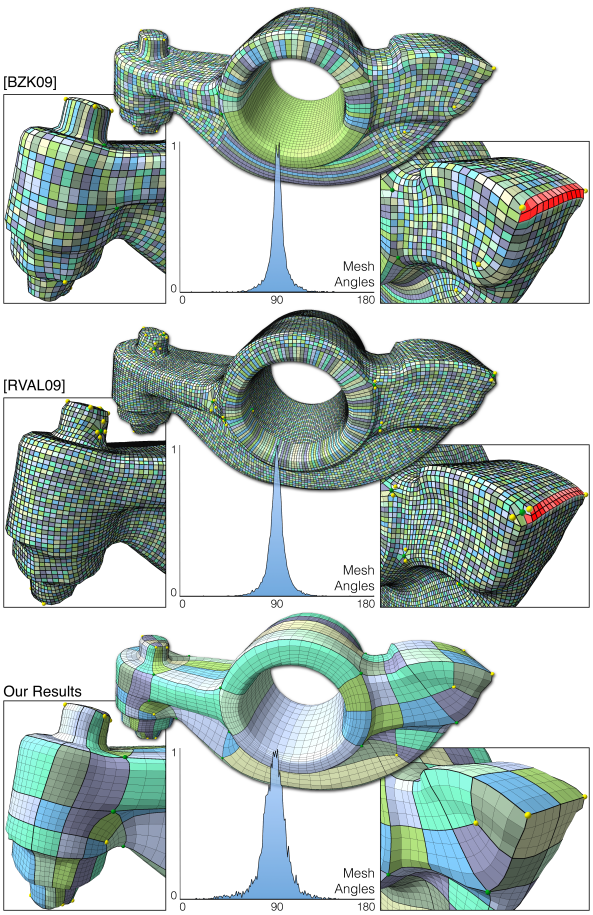


Figure 12: A comparison of the Rocker Arm model between [7], [43] and our approach. At the global scale, the models are decomposed into a coarse set of quads based on the extraordinary vertices. These boundary edges (traced between extraordinary vertices) are shown in bold and demonstrate the structural regularity of our mesh.

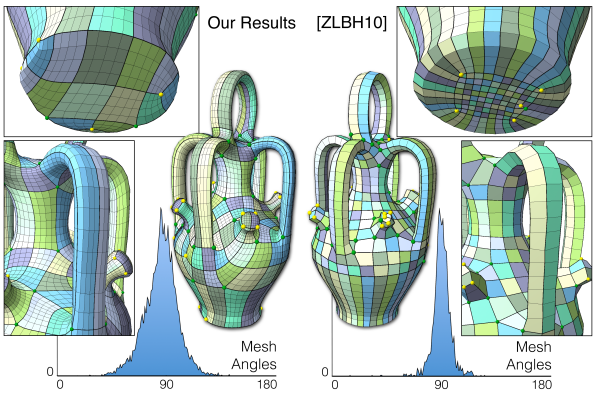


Figure 13: A comparison of the Botijo model between [57] and our approach. We visually compare the structural regularity, decomposing the model into a coarse set of quad regions based on extraordinary vertices. The zoom inset of the base illustrates the coarse boundary edges aligned to feature curves; while the handle inset demonstrates anisotropic sampling.

uolo, M. (Eds.), Shape Analysis and Structuring. Mathematics and Visualization. Springer Berlin Heidelberg.

[3] Attene, M., Falcidieno, B., Rossignac, J., Spagnuolo, M., March 2005. Sharpen-bend: Recovering curved sharp edges in triangle meshes produced by feature-insensitive sampling. IEEE Transactions on Visualization and Computer Graphics 11 (2), 181–192.

[4] Ben-chen, M., Gotsman, C., Bunin, G., 2008. Conformal flattening by curvature prescription and metric scaling. Comp. Graph. Forum 27 (2), 449–458.

[5] Bénière, R., Subsol, G., Puech, W., Gesquière, G., Breton, F. L., 2010. Decomposition of a 3d triangular mesh into quadrangulated patches. In: GRAPP. pp. 96–103.

[6] Boier-Martin, I., Rushmeier, H., Jin, J., 2004. Parameterization of triangle meshes over quadrilateral domains. Symp. on Geom. Proc., 193–203.

[7] Bommes, D., Zimmer, H., Kobbelt, L., 2009. Mixed-integer quadrangulation. ACM Trans. Graph. 28 (3), 1–10.

[8] Carr, N., Hoberock, J., Crane, K., Hart, J., 2006. Rectangular multi-chart geometry images. In: Symp. on Geometry Proc. pp. 181–190.

[9] Chen, Y., Davis, T. A., Hager, W. W., Rajamanickam, S., 2008. Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. ACM Trans. Math. Softw. 35 (3), 1–14.

[10] Cheng, S.-W., Dey, T. K., Ramos, E. A., Ray, T., 2004. Sampling and meshing a surface with guaranteed topology and geometry. In: Proceedings of the Symposium on Computational Geometry. pp. 280–289.

[11] Cohen-Steiner, D., Alliez, P., Desbrun, M., August 2004. Variational shape approximation. ACM Transactions on Graphics 23, 905–914.

[12] Daniels, J., Silva, C., Cohen, E., 2009. Semi-regular quadrilateral-only mesh generation from simplified base domains. Comp. Graph. Forum. 28 (5), 1427–1435.

[13] Davis, T., Hager, W., 2009. Dynamic supernodes in sparse cholesky update/downdate and triangular solves. ACM Transactions Mathematical Software 35 (4), 1–23.

[14] Dong, S., Bremer, P.-T., Garland, M., Pascucci, V., Hart, J., 2006. Spectral surface quadrangulation. ACM Trans. Graph. 25 (3), 1057–1066.

[15] Floater, M. S., 1997. Parametrization and smooth approximation of surface triangulations. Comput. Aided Geom. Des. 14 (3), 231–250.

[16] Fortune, S., 1992. Voronoi diagrams and delaunay triangulation. In: F.K., H., D.Z., D. (Eds.), Comput. in Euclidean Geom. Vol. 1 of Lecture Notes Series on Computing. pp. 193–233.

[17] Gabow, H. N., 1990. Data structures for weighted matching and nearest common ancestors with linking. In: SODA'90. pp. 434–443.

[18] Guskov, I., Khodakovsky, A., Schröder, P., Sweldens, W., 2002. Hybrid meshes: multiresolution using regular and irregular refinement. In: Proceedings of the Symposium on Computational Geometry. pp. 264–272.

[19] Hertzmann, A., Zorin, D., 2000. Illustrating smooth surfaces. In: ACM Trans. Graph. pp. 517–526.

[20] Hildebrandt, K., Polthier, K., Wardetzky, M., 2005. Smooth feature lines on surface meshes. In: Symposium on Geometry Processing. pp. 85–90.

[21] Hormann, K., Polthier, K., Sheffer, A., 2008. Mesh parameterization: theory and practice. In: ACM Trans. Graph. pp. 1–87.

[22] Huang, J., Zhang, M., Ma, J., Liu, X., Kobbelt, L., Bao, H., 2008. Spectral quadrangulation with orientation and alignment control. ACM Trans. Graph. 27 (5), 1–9.
[23] Hubeli, A., Gross, M., 2001. Multiresolution feature extraction for unstructured meshes. In: IEEE Visualization. pp. 287–294.
[24] Kalberer, F., Nieser, M., Polthier, K., 2007. Quadcover: Surface parameterization using branched coverings. Comp. Graph. Forum 26, 375–384.
[25] Khodakovsky, A., Litke, N., Schröder, P., 2003. Globally smooth parameterizations with low distortion. ACM Transactions on Graphics 22, 350–357.
[26] Korte, B., Vygen, J., 2008. Combinatorial Optimization: Theory and Algorithms, 4th Edition. Vol. 21 of Algorithms and Combinatorics. Springer-Verlag, New York, NY, USA.
[27] Kraevoy, V., Sheffer, A., August 2004. Cross-parameterization and compatible remeshing of 3d models. ACM Transactions on Graphics 23, 861–869.
[28] Krishnamurthy, V., Levoy, M., 1996. Fitting smooth surfaces to dense polygon meshes. In: ACM Trans. Graph. pp. 313–324.
[29] Lai, Y.-K., Kobbelt, L., Hu, S.-M., 2008. An incremental approach to feature aligned quad dominant remeshing. In: ACM Symp. on Solid and Phys. Modeling. pp. 137–145.
[30] Lee, A., Sweldens, W., Schroder, P., Cowsar, L., Dobkin, D., 1998. Maps: Multiresolution adaptive parameterization of surfaces. ACM Trans. Graph., 95–104.
[31] Lévy, B., Liu, Y., 2010. Lp centroidal voronoi tessellation and its applications. ACM Trans. on Graph. 29 (4), 1–11.
[32] Lizier, M., Siqueira, M., Daniels, J., Silva, C., Nonato, L., 2010. Template-based remeshing for image decomposition. In: Sibgrapi'10. pp. 1–8.
[33] Losasso, F., Hoppe, H., 2004. Geometry clipmaps: Terrain rendering using nested regular grids. ACM Trans. Graph. 23, 769–776.
[34] Marinov, M., Kobbelt, L., 2004. Direct anisotropic quad-dominant remeshing. In: Pacific Graphics. pp. 207–216.
[35] Myles, A., Pietroni, N., Kovacs, D., Zorin, D., july 2010. Feature-aligned t-meshes. ACM Trans. Graph. 29 (4).
[36] Palacios, J., Zhang, E., July 2007. Rotational symmetry field design on surfaces. ACM Trans. Graph. 26.
[37] Petersen, J., 1891. Die theorie der regulären graphs. Acta Mathematica 15 (1), 193–220.
[38] Pietroni, N., Tarini, M., Cignoni, P., July 2010. Almost isometric mesh parameterization through abstract domains. IEEE Transactions on Visualization and Computer Graphics 16, 621–635.
[39] Pinkall, U., Polthier, K., 1993. Computing discrete minimal surfaces and their conjugates. Exp. Math. 2, 15–36.
[40] Praun, E., Finkelstein, A., Hoppe, H., 2000. Lapped textures. In: ACM Trans. Graph. pp. 465–470.
[41] Ramaswami, S., Ramos, P., Toussaint, G., 1998. Converting triangulations to quadrangulations. Comput. Geom.: Theory and Appl. 9 (4), 257–276.
[42] Ray, N., Li, W. C., Lévy, B., Sheffer, A., Alliez, P., 2006. Periodic global parameterization. ACM Trans. on Graph. 25 (4), 1460–1485.
[43] Ray, N., Vallet, B., Alonso, L., Levy, B., 2009. Geometry-aware direction field processing. ACM Trans. Graph. 29 (1), 1–11.
[44] Schreiner, J., Asirvatham, A., Praun, E., Hoppe, H., August 2004. Inter-surface mapping. ACM Transactions on Graphics 23, 870–877.
[45] Sederberg, T. W., Cardon, D. L., Finnigan, G. T., North, N. S., Zheng, J., Lyche, T., August 2004. T-spline simplification and local refinement. ACM Trans. Graph. 23, 276–283.
[46] Sederberg, T. W., Zheng, J., Bakenov, A., Nasri, A., July 2003. T-splines and t-nurccs. ACM Trans. Graph. 22, 477–484.
[47] Sheffer, A., Praun, E., Rose, K., January 2006. Mesh parameterization methods and their applications. Foundations and Trends in Computer Graphics and Vision 2, 105–171.
[48] Tarini, M., Hormann, K., Cignoni, P., Montani, C., 2004. Polycube-maps. ACM Trans. Graph., 853–860.
[49] Tarini, M., Pietroni, N., Cignoni, P., Panozzo, D., Puppo, E., 2010. Practical Quad Mesh Simplification. Comp. Graph. Forum.
[50] Tong, Y., Alliez, P., Cohen-Steiner, D., Desbrun, M., 2006. Designing quadrangulations with discrete harmonic forms. In: Symp. on Geom. Proc. pp. 201–210.
[51] Velho, L., 2000. Quadrilateral meshing using 4-8 clustering. In: CILANCE 2000. pp. 61–64.
[52] Viswanath, N., Shimada, K., Itoh, T., 2000. Quadrilateral meshing with anisotropy and directionality control via close packing of rectangular cells. In: IMR'00. pp. 227–238.
[53] Wang, H., He, Y., Li, X., Gu, X., Qin, H., 2007. Polycube splines. In: ACM Solid and physical modeling. pp. 241–251.
[54] Xu, K., Cohen-Or, D., Ju, T., Liu, L., Zhang, H., Zhou, S., Xiong, Y., December 2009. Feature-aligned shape texturing. ACM Trans. Graph. 28, 1–7.
[55] Xu, K., Zhang, H., Cohen-Or, D., Xiong, Y., 2009. Dynamic harmonic fields for surface processing. Comput. Graph. 33 (3), 391–398.
[56] Zhang, E., Mischaikow, K., Turk, G., October 2006. Vector field design on surfaces. ACM Trans. Graph. 25, 1294–1326.
[57] Zhang, M., Huang, J., Liu, X., Bao, H., July 2010. A wave-based anisotropic quadrangulation method. ACM Trans. Graph. 29, 118:1–118:8.