Anisotropic Rectangular Metric for Polygonal Surface Remeshing

Bertrand Pellenard[1], Jean-Marie Morvan[2,3], and Pierre Alliez[1]

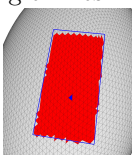
[1] Inria Sophia Antipolis - Méditerranée, France. firstname.lastname@inria.fr
[2] Université Lyon 1/CNRS, France. morvan@math.univ-lyon1.fr
[3] King Abdullah University of Science and Technology, Saudi Arabia.

Summary. We propose a new method for anisotropic polygonal surface remeshing. Our algorithm takes as input a surface triangle mesh. An anisotropic rectangular metric, defined at each triangle facet of the input mesh, is derived from both a user-specified normal-based tolerance error and the requirement to favor rectangle-shaped polygons. Our algorithm uses a greedy optimization procedure that adds, deletes and relocates generators so as to match two criteria related to partitioning and conformity.

1 Introduction

Polygon surface meshes are preferred over triangle meshes in a number of applications related to geometric modeling and reverse engineering. Among those, anisotropic meshes are preferred over isotropic ones when seeking faithful surface approximation for a low number of elements. Inspired by T-meshes [17], the initial motivation of our work is an extension of the variational shape approximation method [7] to generate rectangle-shaped polygons.

In the following we denote by $\mathcal{S}$ the input surface triangle mesh. A *subdomain* is defined as a connected subset of facets of $\mathcal{S}$. A discrete domain *decomposition* of $\mathcal{S}$ is a decomposition of $\mathcal{S}$ into subdomains, possibly with overlaps and orphans. A *generator* is a facet of $\mathcal{S}$ that induces a subdomain through its associated anisotropic rectangular metric (see inset).



Problem statement. Assume an input surface $\mathcal{S}$ given as a surface triangle mesh approximating a piecewise smooth surface, with or without boundaries. We wish to generate a polygon surface mesh where all elements i) are nearly rectangle-shaped, ii) are sized and oriented in accordance to a user-specified tolerance error expressed in local maximum normal deviation to $\mathcal{S}$, and iii) are partitioning $\mathcal{S}$ while favoring conforming configurations where possible, i.e., ideally with no T-junctions. In addition, we wish to minimize

the number of polygons for the specified tolerance. This goal is equivalent to maximize the polygon areas.

While strict local accordance to both sizing and cross fields is already notoriously delicate in the isotropic case, seeking for anisotropic rectangle-shaped polygons with arbitrarily aspect ratios adds other challenges to guarantee island-free and orphan-free partitioning and favor conformity so as to avoid T-junctions [17].

1.1 Previous Work

A common way to state the surface remeshing problem is to say that we wish to decompose the input surface $\mathcal{S}$ with elements that i) partition $\mathcal{S}$ (with no overlaps nor orphans), ii) meet at edges, iii) are well shaped and iv) approximate well $\mathcal{S}$. For specific applications additional criteria may also be sought after, such as regularity, simplicity of the domain structure [3, 22] and more recently, preservation of a remeshing style [23].

Requirements. While the requirements listed above are now relatively well understood for meshing smooth surfaces with isotropic triangle surface meshes, remeshing piecewise smooth surface with anisotropic polygons reveals substantially more difficulty due to more complex requirements that may interact and conflict. We now review each of these requirements:

- Partitioning: decomposing the domain into subdomains with no overlaps nor orphans. In our context each triangle of $\mathcal{S}$ must be covered once.
- Meet at edges: this requirement translates the desire to avoid T-junctions where possible. This requirement conflicts with a graded metric.
- Well shaped: in our context we wish to generate a majority of rectangle-shaped polygons. Note that more basic shape requirements include disc-topology (the later being already not direct for anisotropic partitions [14, 2, 5]) and convexity.
- Approximate well: in our context we wish to approximate well the normals to the input surface $\mathcal{S}$. This translates into control over the (coupled) notions of orientation and size. As we also wish to approximate piecewise smooth surfaces this requires control over alignment, which is not shift invariant contrary to orientation [15]. The approximation requirement, together with the rectangular shape requirement, leads to generate subdomains in accordance to a local anisotropic rectangular metric.

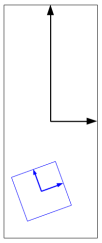
Metric. A common way to compute an anisotropic metric is from curvature estimates [1, 20, 16]. However, this requires choosing a scale such as neighborhood or integration domain, which is not related to an intuitive parameter. In addition, local curvature estimates may lead to incorrect metric such that, e.g., a parabolic zone may dictate an infinite dimension in zero-principal-curvature direction, while the input surface is obviously bounded.

These approaches often require, e.g., bounding anisotropy [16], typically in the range 1 to 5. Furthermore, local curvature tensor estimation is relevant for smooth asymptotic analysis, but anisotropic meshing is often used for generating coarse meshes from piecewise smooth surfaces where the approximation rates drop near sharp creases. Another way to derive an anisotropic metric is to measure normal deviation to the tangent plane [13] or to several tangent planes near sharp features [16]. This is also combined with bounded anisotropy, again in the range 1 to 5 [13].

Remeshing. For quadrangle remeshing much effort has been carried out to control alignment and orientation so as to orient edges to salient directions (often derived from curvature estimates) and align them to features such as sharp creases and boundaries [9, 4]. These approaches construct meshes that are mostly regular, by construction *conforming* and *partitioning* but local sizing is hard to control and hence anisotropy is often not matched. Recent progress has been made on decimation methods [10, 21] but the resulting meshes are still mostly isotropic.

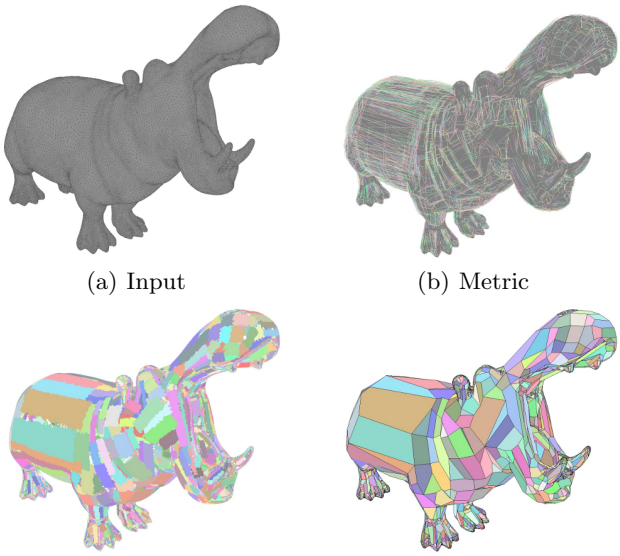
Another way to *guarantee partitioning* by construction consists of constructing centroidal Voronoi diagrams through relaxation with anisotropic L_2 [8] or L_∞ [16] metrics. Conformity is however not optimized for when using the Lloyd iteration as the optimized energy does not relate to conformity. In addition, anisotropy is in general limited to low aspect ratios or to metrics with smooth grading to avoid partitioning defects [5]. Another drawback of such relaxation methods is that control over sizing is only relative such that the optimized cells are sized not strictly in accordance to the input metric but rather proportionally. This requires a special care on providing the relaxation process with the right number of generators. Another relaxation method is the variational shape approximation method [7]. Being solely error-driven the latter is not devised to generate rectangle-shaped polygons and does not optimize for conformity.

A conforming relaxation method has been recently proposed [19] to improve conformity but it also suffers from partitioning defects when employed with strongly anisotropic metrics. The inset depicts a common defect that arises when a generator with a large metric competes with a nearby generator with a small metric. While the method is guaranteed to partition by construction, the local metrics that highly differ may greatly alter the partitioning process and lead to poor accordance to the metric. A better placement algorithm for the generators would not generate such partition defects, but the main issue with relaxation methods is that they rely on such defective partitioning for relocating generators. This suggests that building upon a guaranteed partitioning may not be the best option when seeking meshes with widely anisotropic metrics.



In summary, anisotropic surface remeshing is facing the dilemma between conformity and anisotropy, even more so for rapidly varying metrics where

anisotropy and conformity are highly conflicting. Some algorithms guarantee partitioning and conformity but this often comes at the price of relaxing the metric, either in aspect ratio or in size. Instead of relaxing the metric, we depart from previous work by relaxing partitioning and conformity during the placement of generators and optimize an objective function related to these requirements.



(a) Input (b) Metric

(c) Decomposition (d) Mesh

Fig. 1. Overview. Input surface triangle mesh (a). Anisotropic rectangular metric shown for only 20% of the mesh facets to avoid visual cluttering (b). Decomposition induced by generators after greedy optimization favoring partitioning and conformity (c). Final polygon mesh (d).

1.2 Positioning and Contributions

Our approach bridges the gap between error-driven and metric-driven methods so as to generate anisotropic rectangle-shaped polygons. Our two main contributions are as follows:

1. We derive an anisotropic metric at each facet of the input mesh from both a user-specified maximum tolerance error over the normals, and the requirement to favor rectangle-shaped polygons. In addition to offer arbitrarily aspect ratios, this metric is rectangular and not centered at each facet. Departing from the common approach which uses an elliptic centered metric, such off-centered metric is shown to preserve sharp features.

2. For placing generators we propose a greedy optimization procedure which adds, relocates and deletes generators – in accordance to the absolute metric – so as to optimize for both partitioning and conforming requirements.

2 Overview

The algorithm comprises three steps. We first compute an anisotropic rectangular metric at each facet of the input surface mesh $\mathcal{S} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$. We then generate a decomposition of $\mathcal{S}$ in accordance to the metric through a greedy optimization procedure that optimizes for partitioning and conformity. The final mesh is obtained through discrete Voronoi partitioning followed by a series of local edge collapse operators. The two first steps of the algorithm act on the discrete input surface triangle mesh. The main input parameter is a maximum tolerance error ϵ expressed in local normal deviation to $\mathcal{S}$. Figure 1 depicts the steps of our algorithm, summarized as follows:

Algorithm 1: Anisotropic polygonal surface remeshing Algorithm.

Data: Surface triangle mesh $\mathcal{S}$.
begin
1. Metric (**3**)
2. Decomposition (**4**)
3. Meshing (**5**)

Result: Polygon surface mesh.

3 Metric

An anisotropic rectangular metric is computed at each facet f of $\mathcal{S}$. It is derived from the initial requirement to generate rectangle-shaped subdomains while being close to the input triangle surface mesh $\mathcal{S}$. Close herein means within the maximum tolerance error ($\epsilon > 0$), the latter being expressed in angular deviation to the local normal to $\mathcal{S}$. On defect-laden input meshes we first perform feature-preserving smoothing of the normals [11].

Each facet f of $\mathcal{S}$ is endowed with a canonical unit normal vector $\overrightarrow{n_f}$, compatible with the outward orientation. The deviation between two facets f_1 and f_2 of $\mathcal{S}$ is the (positive) angle $(\overrightarrow{n_{f_1}}\hat{,}\overrightarrow{n_{f_2}})$ between their normals $\overrightarrow{n_{f_1}}$ and $\overrightarrow{n_{f_2}}$, in the range $[0; \pi]$.

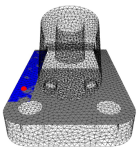
ϵ-tolerance region. Let f be a facet of $\mathcal{S}$. The ϵ-tolerance region $B_{2,1}(f, \epsilon)$ of f is the set of facets of $\mathcal{S}$, connected to f, whose deviation with respect to f is lower than ϵ :

$$B_{2,1}(f,\epsilon) = \{g \in \mathcal{S}, (\overrightarrow{n_g}, \overrightarrow{n_f}) \leq \epsilon\}.$$

This region may be seen as the reciprocal image of a unit ball of radius $\epsilon > 0$ centered at f for the $L_{2,1}$ semi-metric [7]. It may also be related to the mean curvature.

The anisotropic metric $\mathcal{M}_{f,\epsilon}$ of a facet f with tolerance ϵ is the rectangle constructed as follows:

- Let $\overrightarrow{\mathbf{n}}$ be the average normal vector of $B_{2,1}(f,\epsilon)$.
- Let $\overrightarrow{\mathbf{n}}^{\perp}$ be the (affine) 2-plane orthogonal to $\overrightarrow{\mathbf{n}}$ add passing through the center of f : $c(f)$.
- Let $\Pi_{\overrightarrow{\mathbf{n}}^{\perp}}(B_{2,1}(f,\epsilon))$ be the orthogonal projection of $B_{2,1}(f,\epsilon)$ on $\overrightarrow{\mathbf{n}}^{\perp}$.



The anisotropic rectangular metric $\mathcal{M}_{f,\epsilon}$ is the largest rectangle inscribed in $\Pi_{\overrightarrow{\mathbf{n}}^{\perp}}(B_{2,1}(f,\epsilon))$ containing $\Pi_{\overrightarrow{\mathbf{n}}^{\perp}}(c(f))$. The key justification for choosing the largest rectangle inscribed in $\Pi_{\overrightarrow{\mathbf{n}}^{\perp}}(B_{2,1}(f,\epsilon))$ containing $\Pi_{\overrightarrow{\mathbf{n}}^{\perp}}(c(f))$ is that the tolerance region is in general not rectangle-shaped. The inset depicts the largest rectangle (blue) inscribed in the tolerance region (gray) containing the query facet (red). We extended an existing algorithm for computing the largest axis-aligned rectangle empty of query points [18] such that the largest rectangle is constrained to contain a given query point (here $c(f)$), and is not necessarily axis-aligned through a simple uniform sampling of the directions. Figure 2 depicts the anisotropic rectangular metric evaluated at a single facet for both smooth and piecewise smooth shapes.

Figure 3 depicts the set of shift vectors for all facets of $\mathcal{S}$. Note that on the 6 faces of a parallelepiped all shift vectors point to the center of the face, which is intrinsically redundant. We explain next how our greedy optimization procedure takes advantage of such redundancy to pick the best generator among the many degrees of freedom offered by this redundancy. Note also that near sharp creases the usual anisotropic metric described as an ellipse or rectangle centered at the query facet would not preserve the crease.

We now define some notions behind the anisotropic rectangular metric, the L_∞ distance in the anisotropic metric and the corners for each metric.

Metric. The anisotropic rectangular metric of facet f, denoted by $\mathcal{M}_{f,\epsilon}$, is defined by a shift vector from $c(f)$ to $\Gamma(f)$ (the rectangle center, in general not inside f) and two non-unit orthogonal vectors $\vec{u_f}$ and $\vec{v_f}$.

L_∞ distance. The L_∞ distance between p and q in the metric $\mathcal{M}_{f,\epsilon}$ is defined as follows :

$$d_\infty^{\mathcal{M}_{f,\epsilon}}(q,p) = \max\left\{\frac{|\vec{pq}\cdot\vec{u_f}|}{||\vec{u_f}||^2}, \frac{|\vec{pq}\cdot\vec{v_f}|}{||\vec{v_f}||^2}\right\},$$

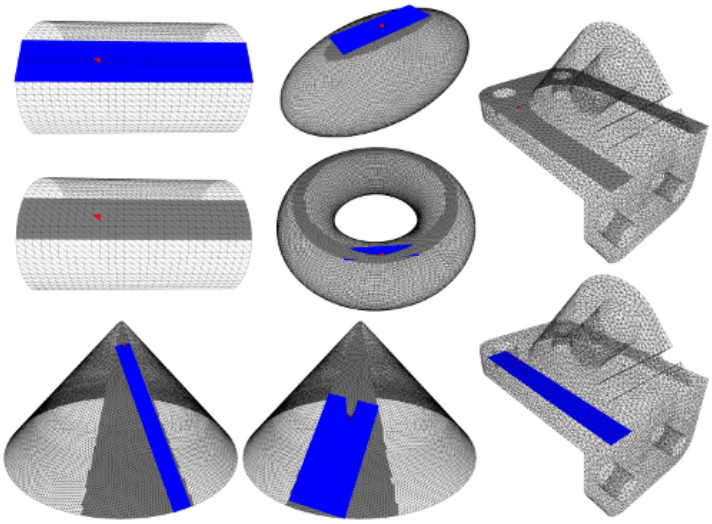


Fig. 2. Example metrics shown for a single query facet. We first compute the tolerance region (gray) for the query facet (red), then compute the largest inscribed rectangle (blue) in this tolerance region, that contains the facet. On these examples ϵ is set to 15 degrees. Note how the tolerance region can be far from being rectangular (elliptic for ellipsoid, ring for the torus, triangle for the cone, u for the anchor), and the metric can be far from being centered at the query facet. On the cone the metric varies in shape and area depending on the location of the query facet.

where $\cdot$ denotes the usual dot product in $\mathbb{R}^3$. In order to take into account the shift vector of a metric, the L_∞ distance of a facet h from the center of the metric $\mathcal{M}_{f,\epsilon}$ is given by $d_\infty^{\mathcal{M}_{f,\epsilon}}(c(h), \Gamma(f))$, where $c(h)$ denotes the centroid of facet h.

Metric corner vertices. For each facet f, the corner vertices of its metric $\mathcal{M}_{f,\epsilon}$ are defined as the four vertices of $\mathcal{S}$ which are respectively the closest from the four corner points of its unit rectangle in the local metric. These four corner points have as approximate L_∞ distance coordinates $(-1,-1)$, $(-1,1)$, $(1,-1)$ and $(1,1)$ in metric $\mathcal{M}_{f,\epsilon}$.

Figure 4 depicts the anisotropic rectangular metric evaluated at some facets of the hippo model. The metric can be locally redundant (if several shift vectors point to the same area), and conflicting both in size and in orientation.

4 Decomposition

We now wish to generate a decomposition in best accordance to the precomputed anisotropic rectangular metric while matching the requirements listed

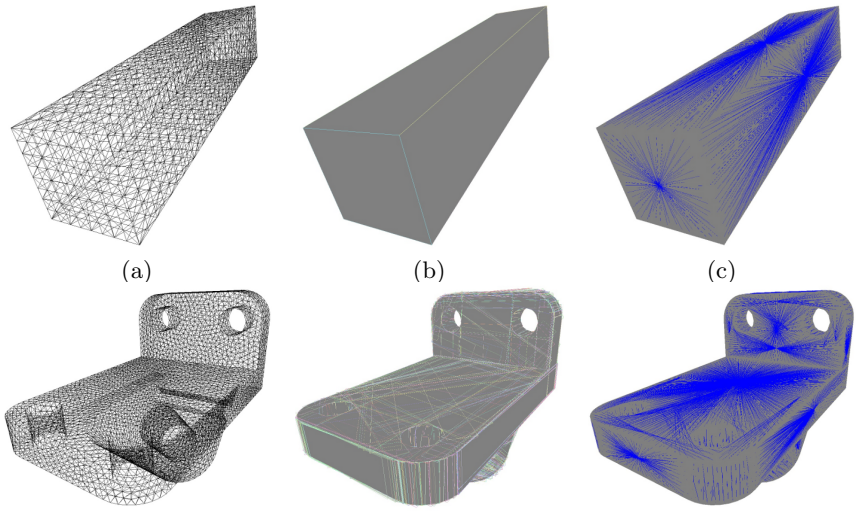


Fig. 3. For each input mesh (a),(d), we show some metrics (b), (e) and draw all shift vectors between a facet and its associated metric center (c),(f).

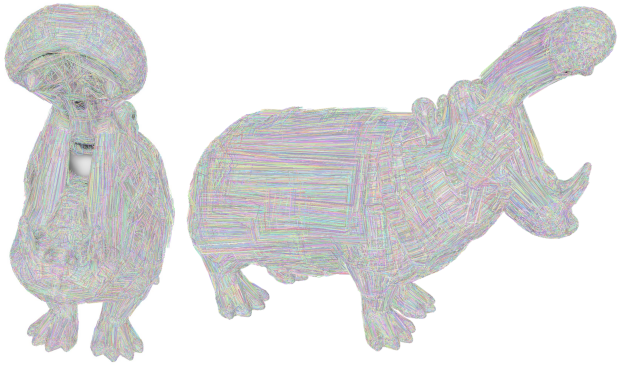


Fig. 4. Metric shown for only 10% of the facets to avoid visual cluttering.

in Section 1. The decomposition is obtained by placing generators through a greedy optimization procedure that optimizes for partitioning and conformity.

At each facet f of the input surface mesh $\mathcal{S}$ is associated a *generator* with the corresponding anisotropic rectangular metric. Local metrics may greatly overlap and conflict. Our goal bears some similarity with the set-cover problem as we search for the minimum set $\mathcal{G}$ of generators such that the decomposition best matches the requirements. Remind that the general set-cover problem, shown to be NP-hard, consists in finding the minimum number of subsets of a set that cover the input domain [12]. Instead of guaranteeing partitioning by construction when using, e.g., L_∞ Voronoi diagrams [16, 19], we favor

during optimization the absolute matching of the metric while allowing a decomposition with both overlaps and orphans. In other words, the partitioning requirement now becomes an objective function to optimize through the notion of *coverage*.

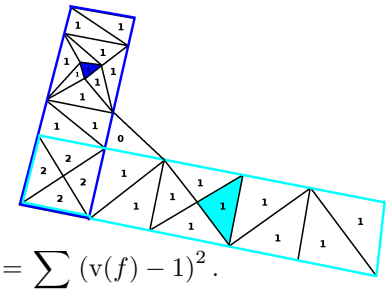
Coverage. We define a coverage energy which measures how the current generators satisfy the requirement to cover each facet of $\mathcal{S}$ once. We first define the covering set $\mathcal{C}_{\mathcal{M}_{g,\epsilon}}$ of a generator g as the connected set of facets containing g, within the ϵ-tolerance region of g and whose L_∞ distance in the metric $\mathcal{M}_{g,\epsilon}$ to g is lower than 1:

$$\mathcal{C}_{\mathcal{M}_{g,\epsilon}} = B_{2,1}(g,\epsilon) \bigcap \left\{ f \in \mathcal{F} : d_\infty^{\mathcal{M}_{g,\epsilon}}(c(f), \Gamma(g)) \leq 1 \right\}.$$

In addition, we define for each facet f of $\mathcal{S}$ its coverage $\mathrm{v}(f)$ as the number of generators that cover f:

$$\mathrm{v}(f) = \{ g \in \mathcal{G} : f \in \mathcal{C}_{\mathcal{M}_{g,\epsilon}} \}.$$

The inset depicts the facet coverage when placing two generators.



The total coverage energy is defined as: $E_\mathrm{v} = \sum_{f \in \mathcal{F}} \left(\mathrm{v}(f) - 1 \right)^2$.

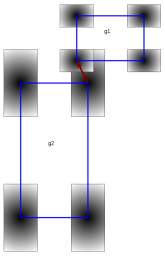
Conformity. A *subdomain corner* is a vertex of $\mathcal{S}$ adjacent to three subdomains or more in the interior, or to two or more subdomains on the boundary of $\mathcal{S}$. A decomposition is said conforming when the domain decomposition contains no T-junctions. In our setup a relaxed version of this notion means that all subdomain corners nearly coincide with metric corner vertices (defined above). We define a conformity energy which favors close metric corners for all generated subdomains. This energy is computed for each vertex of $\mathcal{S}$ and depends on the generators placed in its neighborhood. The latter requires defining the coverage of a vertex as the set of generators that cover it according to a loose L_∞ distance (set to 2). We denote this coverage by $\mathcal{N}_v$. The conformity energy of a vertex is defined as :

$$E_\mathrm{f}(v) = \frac{1}{\binom{|\mathcal{N}_v|}{2}} \sum_{\substack{g_1, g_2 \in \mathcal{N}_v \\ g_1 \neq g_2}} ||\chi_{v,g_1} - \chi_{v,g_2}||_2 \left(\frac{\Phi_{g_1}(\chi_{v,g_1}, v)}{\bar{g_1}} + \frac{\Phi_{g_2}(\chi_{v,g_2}, v)}{\bar{g_2}} \right),$$

where $\chi_{v,g}$ denotes the closest corner of g from v (in the metric of g), $\bar{g}$ denotes the average norms of the two metric vectors of g (such that the distance between two interacting corners is normalized to the local metrics), $\binom{|\mathcal{N}_v|}{2}$ denotes the number of oriented pairs (g_1, g_2), and

$$\Phi_g(p,q) = 4.\mathbb{1}_{\{(p,q): d_\infty^{\mathcal{M}_{g,\epsilon}}(p,q) \leq \frac{1}{2}\}} \left(\frac{1}{2} - \frac{|\overrightarrow{pq} \cdot \overrightarrow{u_g}|}{||\overrightarrow{u_g}||^2} \right) \left(\frac{1}{2} - \frac{|\overrightarrow{pq} \cdot \overrightarrow{v_g}|}{||\overrightarrow{v_g}||^2} \right),$$

where $\mathbb{1}$ is the indicatrix function, and $\overrightarrow{u_g}$ and $\overrightarrow{v_g}$ are the two metric vectors of g. $\Phi_g(p,q)$ is an importance function valued 1 at each subdomain corner



that decreases to zero with bilinear interpolation at the L_∞ distance 0.5 (see inset: from white to black by increasing importance). The rationale behind this formula is to penalize distant corners between all oriented pairs of neighboring subdomains, for all vertices of $\mathcal{S}$, and with a higher importance coefficient for vertices near the metric corners. Choosing oriented instead of unoriented pairs allows us to measure the distance between corners in the metric of both g_1 and g_2.

As defined above the conformity energy $E_\mathrm{f}(v)$ is non-zero only if at least two subdomains interact, i.e., overlap according to their 2-L_∞ distance. As we wish to penalize the generation of isolated subdomain we add a special case when a subdomain does not interact with any subdomain. In this case we set $E_\mathrm{f}(v) = 1$, which amounts to consider a highly penalizing corner distance. During the first phase of the optimization procedure – which mostly adds generators – this tends to favor an advancing front instead of a random placement of isolated subdomains, the latter revealing detrimental to conformity. The total conformity energy is defined as: $E_\mathrm{f} = \sum_{v \in \mathcal{V}} E_\mathrm{f}(v)$.

Global energy. The global optimized energy has terms on facets of $\mathcal{S}$ and on vertices of $\mathcal{S}$: $E = E_\mathrm{v} + \alpha E_\mathrm{f}$, where α is used to trade conformity for partitioning (set to 0.5 in all examples shown).

Energy minimization. The global energy is minimized through a greedy optimization procedure which adds, relocates and deletes generators, such that the corresponding overlapping decomposition is optimized for partitioning and conformity. More specifically, we use a modifiable priority queue with polymorphic operators (addition, relocation, deletion), sorted by decreasing order of energy changes, referred to as the operator cost. While the addition and deletion operators are defined per generator, the relocation operator is defined for each feasible pair of generators. To avoid dealing with a large combinatorial complexity we thus restrict relocation of each generator g to a random subset (set to 20% in all examples shown) of the (unused) generators only among the ones covered by g. During optimization, and to efficiently update the priority queue, we dynamically maintain for each generator the list of generators covered by it as well as the list of generators that cover it. We perform the same book-keeping for vertices but with a looser notion of covering: the 2-L_∞ distance with respect to the metric of each generator.

Initially we insert no generators on $\mathcal{S}$, and the priority queue is initialized with only addition operators that correspond to all generators (as many as the number of facets of $\mathcal{S}$). At each step of the algorithm (an operator popped out of the queue) we update the priority queue by removing unfeasible operators, updating the cost of the modified feasible operators, and adding all new feasible operators. The algorithm stops when no decrease of the energy is possible. Algorithm 2 summarizes the optimization process.

Algorithm 2: Greedy optimization

Data: Triangle surface mesh $\mathcal{S}$.
Result: Set of generators $\mathcal{G}$ on $\mathcal{S}$.
begin
  $\mathcal{G} = \emptyset$;
  Enumerate $\mathcal{C}$ operators (addition, deletion, relocation);
  Fill priority queue with $\mathcal{C}$;
  while *next operator in queue decreases energy* **do**
    Pop operator from queue;
    Apply operator;
    Update $\mathcal{G}$;
    Update queue;

As expected this greedy algorithm starts by a vast majority of addition operators (with the largest subdomains first to improve coverage), before automatically switching to a relaxation phase where a large majority of relocation operators are chosen. Figure 5 depicts the energy decrease during optimization. The curves shows a faster decrease of the coverage than of the conformity term of the energy. Most relocation operators occur toward the end. Only a few deletion operators occur especially on isotropic areas where the metric is rapidly varying in direction but not in size.

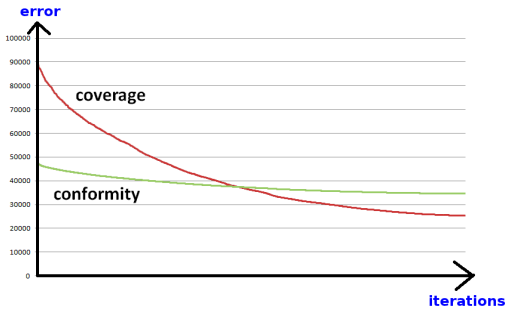


Fig. 5. Energy decrease. The curves show the coverage and the conformity components of the energy. The iterations on the horizontal axis corresponds to the number of operators popped out of the priority queue.

Figure 6 illustrates the role of the conformity term in the energy. The subdomains are not perfectly conforming but conformity is largely improved where the metric offers enough degrees of freedom among all generators.

Figure 7 illustrates a little more subtle behavior of the conformity term of the energy. On this isotropic example all metrics are equally sized but there is no favored direction due to isotropy. This makes the metric highly incompatible in orientation. When running the greedy optimization without

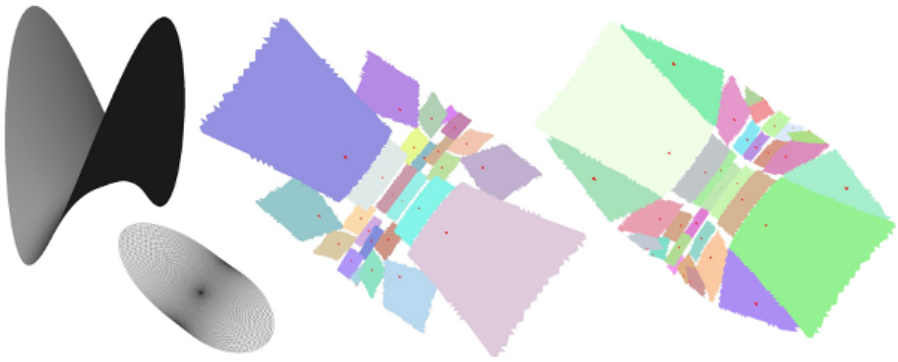


Fig. 6. Optimizing for conformity. On this anisotropic saddle (shown left) we compare the placement of generators without (middle) and with (right) the conformity term in the energy. The saddle is shown from above.

the conformity term the algorithm favors only coverage and hence pops addition operators in decreasing order of area without choosing locally compatible directions. When running the optimization with the conformity term the algorithm pops operators in an order which balances coverage and conformity. This induces a decomposition locally more compatible in orientation as the priority can pick a good subset of generators among all feasible generators.

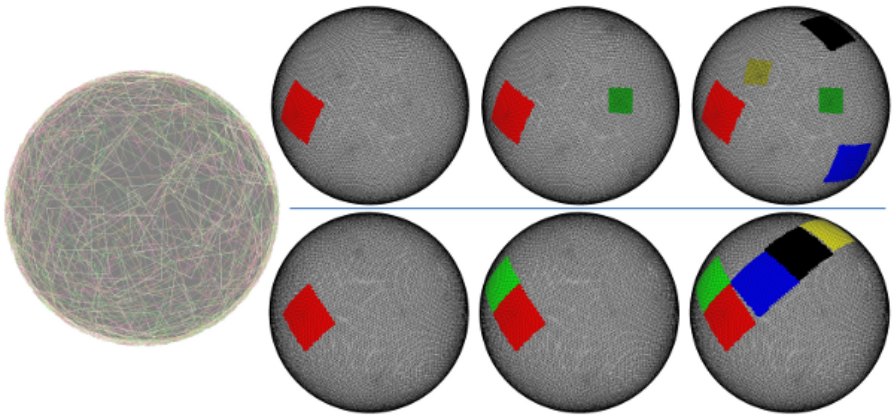


Fig. 7. Placement on a sphere. Left: the metric is uniformly sized but highly not compatible in orientation. Top: decomposition without conformity term in the energy. Bottom: decomposition with the conformity term.

5 Meshing

From the current decomposition we now wish to generate a polygon surface mesh. Note that the greedy optimization procedure optimizes for it, but does not guarantee the partitioning requirement: generally, $E_v \neq 0$. We first generate a partition of $\mathcal{S}$ through a discrete Voronoi partitioning computed over the triangles of $\mathcal{S}$. We then generate a polygon surface mesh, denoted by $\mathcal{M}$, induced by this partition. As post-processing we then apply a series of local edge collapse operators to $\mathcal{M}$ in order to eliminate some of the T-junctions (Figure 8).

Discrete Voronoi Partitioning. Similarly to previous work [19, 7] we generate a partition by flooding the input mesh $\mathcal{S}$ one triangle at a time from the set of generators placed during decomposition. We use a global priority queue initialized with all feasible triangles incident to the generators. The triangles are popped out of the queue and added to the partition in increasing order of anisotropic distance to the off-center of their respective generator. Each distance is measured with respect to its respective metric.

Each vertex of $\mathcal{S}$ is now located either at the interior of a partition subdomain or at the interface between 2 or more partition subdomains. A vertex is said of degree n (with $n > 1$) if it is adjacent to n subdomains. Following the terminology of Pellenard et al. [19], we tag as meta-vertices the vertices of $\mathcal{S}$ with degree ≥ 3 in the interior, and ≥ 2 on the boundary of $\mathcal{S}$. We then generate the polygon surface mesh $\mathcal{M}$ induced by the partition and its meta-vertices.

Local operators. Note that the greedy optimization procedure optimizes for it, but does not guarantee the conformity requirement. In some places $\mathcal{M}$ thus contains nearby T-junctions connected by short edges which can be easily eliminated through edge collapse operators. Using a priority queue we thus recursively collapse short edges by increasing length, where the length is expressed in the local anisotropic metric. The length of short edges is limited to a small fraction of the unit local length (set to $\beta = 0.25$ in all experiments shown).

6 Results

The parameters of our algorithm are summarized as follows:

- ϵ: tolerance error
- α: trade coverage for conformity (Section 4)
- β: maximum length of collapsed edges (Section 5)

Our algorithm is implemented in C++ using the CGAL library [6]. All examples are computed on a PC with 4 cores clocked at 2.40GHz. Computational times are in the order of 30mn for pre-computing the anisotropic metric

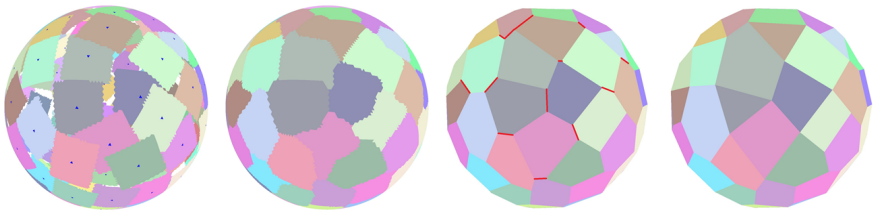
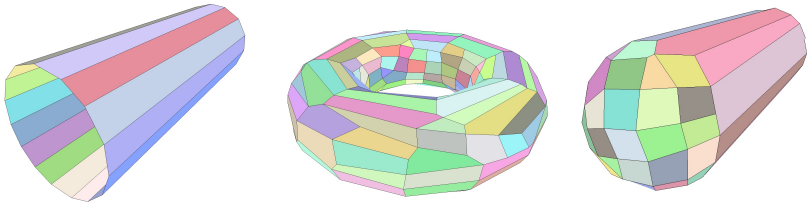


Fig. 8. Meshing. Generators and induced decomposition with overlaps and orphans (left). Partition after discrete Voronoi partitioning (middle left). Induced polygon mesh with short edges depicted in red (middle right). Final polygon mesh after collapsing short edges.

and 40mn for placing generators on an input mesh with $200K$ triangles (resp. 5 and 2.5 minutes for a mesh with 20K triangles). During the computation of the metric 95% of the time is spent at finding the largest rectangles. During the placement of generators 90% of the time is spent at measuring the conformity energy.

Figure 9 is a sanity check on canonical shapes. On the cylinder the polygons have an aspect ratio close to 10. Aspect ratios of 100 are obtained when using a smaller normal tolerance error in combination with a finer input mesh. On the torus the metric varies in size and thus the final meshing conforms only where possible. On the capsule the spherical caps conform where possible to the parabolic area. The conformity term of the energy favors one arbitrary direction on the spherical caps.



(a) Cylinder (b) Torus (c) Capsule

Fig. 9. Remeshing for canonical shapes.

Figure 10 illustrates the behavior of our algorithm on the Fandisk model where the rectangular metric highly conflicts on surface patches with concave boundaries.

Figure 11 illustrates the behavior of our algorithm when varying the normal tolerance error on the ellipsoid.

Figure 12 compares our approach with VSA [7]. Our method behaves well even on nearly planar areas as well as on the turning parabolic areas such as the arms. VSA generates during the optimization phase a partition which is

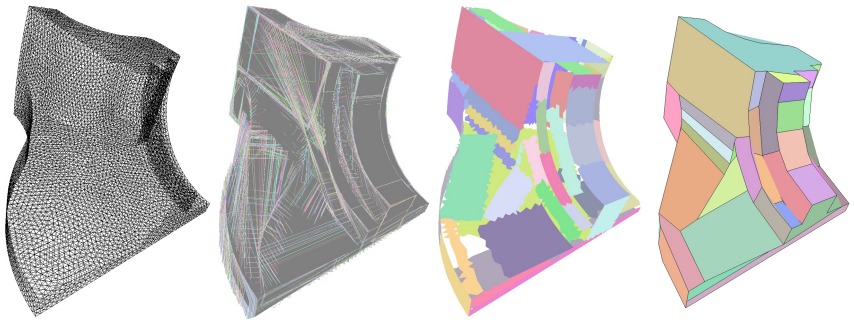
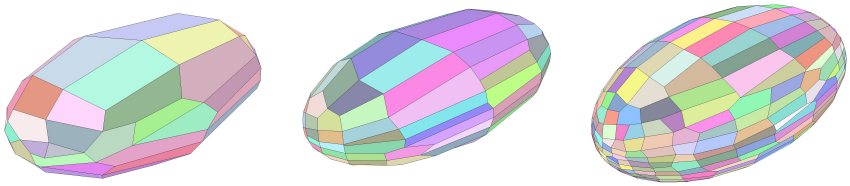


Fig. 10. Fandisk. Top: input mesh, metric and decomposition by generators placed. Bottom: partition defect, partition and final polygon mesh.



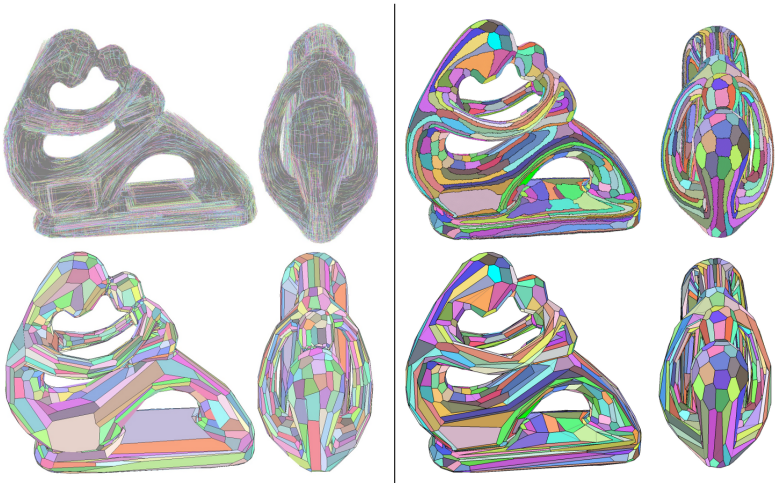
(a) $\epsilon = 15$ degrees (b) $\epsilon = 10$ degrees (c) $\epsilon = 6$ degrees

Fig. 11. Ellipsoid. Remeshing for decreasing normal tolerance errors: 15 degrees (a), 10 degrees (b) and 6 degrees (c).

hampered with noise on nearly planar areas, and composed of curvy regions on the arms which are far from being convex and hence challenges the meshing step.

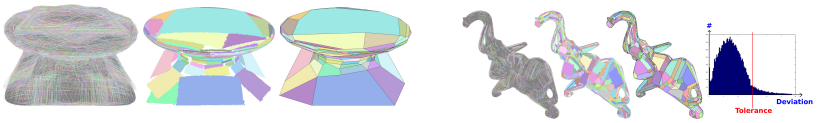
Figure 13(b) illustrates our approach on the elephant model. Figure 13(a) illustrates a case of rapidly varying metric. The distribution of angular deviations depicted by Figure 13(b) shows a rapid drop beyond the user-specified tolerance error.

Limitations. One limitation of our approach is the computational time to compute the metric (in the order of several minutes for 30K vertices), with most of the time spent at computing the largest inscribed rectangle. Computing the conformity term of the energy during optimization is also labor-intensive, mostly due to the time and book-keeping required to detect the neighboring subdomains of each vertex. Another limitation lies into the fact that the orphan parts of the input mesh (not covered by generators during decomposition) which are partitioned during the discrete Voronoi partitioning step may lead to concave polygons. Finally, our greedy optimization performs one operator at a time, and thus often finds local minima except for simple canonical shapes.



(a) Our method (b) VSA

Fig. 12. Comparison with VSA [7].



(a) Mushroom (b) Elephant

Fig. 13. Mushroom and Elephant. Left: metric, decomposition and mesh for mushroom. Right: metric, decomposition, mesh, and distributions of angular deviation between the input and the final meshes.

7 Conclusion

We introduced a novel feature-preserving anisotropic rectangular metric based on finding the largest inscribed rectangle inside a normal-based local tolerance region. This metric is defined at each facet of the input surface mesh by a rectangle and a shift vector which helps preserving features and generating large polygons. We proposed a simple greedy optimization procedure for placing generators which departs from previous work by using an overlapping decomposition instead of a partition, and by simultaneously optimizing for partitioning and conformity.

The added value of our approach is its capability to generate polygonal meshes where the polygons are nearly rectangle-shaped with arbitrary anisotropic aspect ratios. Our domain decomposition algorithm proceeds in absolute accordance to the metric, and automatically switches from refinement to relaxation.

As future work we wish to investigate how the vector field formed by the metric shift vectors can be analyzed to accelerate the placement of genera-

tors. We also wish to use not just largest rectangles but also largest triangles and ellipses so as to adjust the degree of the polygons to the local surface geometry. We also want to find a way to process the metric through, e.g., quantization, so as to further reinforce conformity in cases where strict accordance to the metric is not a priority. Finally, we will investigate a dynamic programming approach capable of placing more than one generator at a time during optimization so as to reach lower minima of the energy.

Acknowledgments

We wish to thank Julie Digne and David Bommes for feedback. This work was funded by the European Research Council (ERC Starting Grant 'Robust Geometry Processing', Grant agreement 257474), and by a French ANR Grant (GIGA ANR-09-BLAN-0331-01).

References

1. P. Alliez, D. Cohen-Steiner, O. Devillers, B. Lévy, and M. Desbrun. Anisotropic polygonal remeshing. *ACM Trans. Graph.*, 22(3):485–493, 2003.
2. J.-D. Boissonnat, C. Wormser, and M. Yvinec. Locally uniform anisotropic meshing. In *Proc. Annual Symposium on Computational Geometry*, pages 270–277, 2008.
3. D. Bommes, T. Lempfer, and L. Kobbelt. Global structure optimization of quadrilateral meshes. *Comput. Graph. Forum*, 30(2):375–384, 2011.
4. D. Bommes, H. Zimmer, and L. Kobbelt. Mixed-integer quadrangulation. *ACM Trans. Graph.*, 28(3), 2009.
5. G. D. Canas and S. J. Gortler. Orphan-free anisotropic voronoi diagrams. *Discrete Comput. Geom.*, 46:526–541, 2011.
6. CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.
7. D. Cohen-Steiner, P. Alliez, and M. Desbrun. Variational shape approximation. *ACM Trans. Graph.*, 23:905–914, 2004.
8. Q. Du and D. Wang. Anisotropic centroidal voronoi tessellations and their applications. *SIAM J. Scientific Computing*, 26(3):737–761, 2005.
9. J. Huang, M. Zhang, J. Ma, X. Liu, L. Kobbelt, and H. Bao. Spectral quadrangulation with orientation and alignment control. In *ACM SIGGRAPH Asia 2008 papers*, SIGGRAPH Asia '08, pages 147:1–147:9, 2008.
10. J. Daniels II, C. T. Silva, and E. Cohen. Localized quadrilateral coarsening. *Comput. Graph. Forum*, 28(5):1437–1444, 2009.
11. T. R. Jones, F. Durand, and M. Desbrun. Non-iterative, feature-preserving mesh smoothing. *ACM Transactions on Graphics*, 22(3), 2003.
12. R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972.
13. D. Kovacs, A. Myles, and D. Zorin. Anisotropic quadrangulation. In *Symposium on Solid and Physical Modeling*, pages 137–146, 2010.
14. F. Labelle and J. R. Shewchuk. Anisotropic voronoi diagrams and guaranteed-quality anisotropic mesh generation. In *Proc. Annual Symposium on Computational Geometry*, pages 191–200, 2003.

15. Y.-K. Lai, L. Kobbelt, and S.-M. Hu. An incremental approach to feature aligned quad dominant remeshing. In *Proceedings of the ACM symposium on Solid and physical modeling*, SPM '08, pages 137–145, 2008.
16. B. Lévy and Y. Liu. L^p centroidal voronoi tessellation and its applications. *ACM Transactions on Graphics*, 29(4), 2010.
17. A. Myles, N. Pietroni, D. Kovacs, and D. Zorin. Feature-aligned T-meshes. *ACM Trans. Graph*, 29(4), 2010.
18. M. Orlowski. A new algorithm for the largest empty rectangle problem. *Algorithmica*, 5(1):65–73, 1990.
19. B. Pellenard, P. Alliez, and J.-M. Morvan. Isotropic 2d quadrangle meshing with size and orientation control. In *Proceedings of the International Meshing Roundtable*, pages 81–98, 2011.
20. N. Ray, W. C. Li, B. Lévy, A. Sheffer, and P. Alliez. Periodic global parameterization. *ACM Trans. Graph.*, 25(4):1460–1485, 2006.
21. M. Tarini, N. Pietroni, P. Cignoni, D. Panozzo, and E. Puppo. Practical quad mesh simplification. *Comput. Graph. Forum*, 29(2):407–418, 2010.
22. M. Tarini, E. Puppo, D. Panozzo, N. Pietroni, and P. Cignoni. Simple quad domains for field aligned mesh parametrization. *ACM Transactions on Graphics*, 30(6), 2011.
23. J. Tierny, J. Daniels II, L. Gustavo Nonato, V. Pascucci, and C. T. Silva. Inspired quadrangulation. *Computer-Aided Design*, 43(11):1516–1526, 2011.