Anisotropic Delaunay meshes of surfaces.
Jean-Daniel Boissonnat, Kan-Le Shi, Jane Tournois, Mariette Yvinec

To cite this version:
Jean-Daniel Boissonnat, Kan-Le Shi, Jane Tournois, Mariette Yvinec. Anisotropic Delaunay meshes of surfaces.. ACM Transactions on Graphics, Association for Computing Machinery, 2014. <hal-00920678>

HAL Id: hal-00920678
https://hal.inria.fr/hal-00920678
Submitted on 19 Dec 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Anisotropic Delaunay Meshes of Surfaces

Jean-Daniel Boissonnat
INRIA Sophia-Antipolis
and
Kan-Le Shi
Tsinghua University
and
Jane Tournois
GeometryFactory
and
Mariette Yvinec
INRIA Sophia-Antipolis

Anisotropic simplicial meshes are triangulations with elements elongated along prescribed directions. Anisotropic meshes have been shown to be well suited for interpolation of functions or solving PDEs. They can also significantly enhance the accuracy of a surface representation. Given a surface $S$ endowed with a metric tensor field, we propose a new approach to generate an anisotropic mesh that approximates $S$ with elements shaped according to the metric field. The algorithm relies on the well-established concepts of restricted Delaunay triangulation and Delaunay refinement and comes with theoretical guarantees. The star of each vertex in the output mesh is Delaunay for the metric attached to this vertex and the facets have good aspect ratio with respect to this metric. The algorithm is easy to implement. It can mesh various types of surfaces like implicit surfaces, polyhedra or isosurfaces in 3D images. It can handle complicated geometries and topologies, and very anisotropic metric fields.

Categories and Subject Descriptors: I.3.5 [Computing Methodologies]: Computer Graphics—Geometry and Object Modeling

General Terms: Algorithms

1. INTRODUCTION

Anisotropic simplicial meshes are triangulations with elements elongated along prescribed directions. Anisotropic meshes have been shown to be particularly well suited for interpolation of functions and solving PDE’s [D’Azevedo and Simpson 1989; Shewchuk 2002; Mirebeau 2010]. They can also significantly enhance the accuracy of a surface representation if the anisotropy of the mesh conforms to the curvature tensor of the surface [Heckbert and Garland 1999]. In this paper, we consider the problem of sampling and meshing a surface $S$ endowed with a metric tensor field that varies over $S$. The metric tensor can be chosen arbitrarily provided that it satisfies a continuity condition over the surface $S$. The metric tensor may, for instance, be the curvature tensor of $S$ or the Hessian matrix of a function defined over $S$.

1.1 Background

Due to its practical importance, several methods have been proposed in the literature to address the problem of approximating surfaces using anisotropic surface meshes. The case of a parametric surface $S$ can be handled by working in the 2-dimensional space $U$ of the parameters of $S$, using an appropriate metric field to account for the deformation induced by the mapping $U \rightarrow S$. The problem is then 2-dimensional [Borouchaki et al. 1997; Bossen and Heckbert 1996; Li et al. 1999]. Several methods have also been reported to remesh a polyhedral surface according to some specified metric tensor. Alliez et al. [Alliez et al. 2003] first estimate the principal direction fields and trace a network of lines of curvature from which they deduce a quad-dominant anisotropic mesh. Jiao et al. [Jiao et al. 2006] use the quadratic metric tensor of Heckbert and Garland [Heckbert and Garland 1999] and propose a set of operations to allow anisotropic adaptation of a triangular mesh.
to static or evolving surfaces. The case of implicit surfaces can be handled by first polygonizing the surface in a standard way and then remeshing the obtained surface using one of the anisotropic methods mentioned above. Although this approach is clearly not optimal, very few methods have been proposed to directly generate anisotropic meshes on implicit surfaces. One exception is the work of Azenkov and Fischer [Azenkov and Fischer 2005] who propose a grid-based approach to mesh implicit surfaces according to their curvature tensor. The method uses a deformed grid and produces quad-meshes. The anisotropic grid is obtained by deforming a cartesian grid while maintaining the structured topology, which is a limitation when considering complex shapes.

In order to settle more rigorous foundations and to precisely characterize anisotropic triangulations, several authors have introduced the anisotropic Voronoi diagrams (AVD). An anisotropic triangulation, called anisotropic Delaunay triangulation (ADT), is then defined as the dual of the AVD. Various AVD and ADT have been proposed. Leibon and Letscher [Leibon and Letscher 2000] introduced Delaunay triangulations and Voronoi diagrams for Riemannian manifolds. This approach requires to compute geodesic paths and intrinsic balls, which may be quite complicated in practice. To overcome this issue, two approaches have been proposed that approximate the geodesic distance between a given site and any point by considering the metric constant and equal to the metric at the site [Labelle and Shewchuk 2003] or at the point [Du and Wang 2005; Lévy and Liu 2010]. Although nice results have been reported, it is recognized in [Du and Wang 2005] that “more precise theoretical analysis are needed in identifying conditions on the metric, geometry, and the generator distributions that guarantee the well-defined duality between the AVDs and the ADTs”. The Labelle and Shewchuk approach [Labelle and Shewchuk 2003] is easier to analyze and, in the 2-dimensional case, the authors have proposed a refinement algorithm that can provably produce anisotropic meshes. This approach has been extended by Cheng et al. [Cheng et al. 2006] to produce anisotropic meshes of surfaces embedded in \( \mathbb{R}^3 \). The algorithm is however very complicated and no implementation has been reported. Although some progress has been recently reported in [Canas and Gortler 2011], it remains unclear under which conditions an anisotropic Voronoi diagram admits a dual embedded triangulation for 3 or higher dimensional domains.

A different approach has been proposed by Boissonnat, Wormser and Yvinec [Boissonnat et al. 2008]. They introduced a new variant of anisotropic Delaunay meshes, called locally uniform anisotropic meshes. Such meshes are not defined by duality from some anisotropic Voronoi diagram. Instead, it is defined as a simplicial mesh in which the star of each vertex is Delaunay for the metric attached to the vertex. It is shown that, when the metric field satisfies some continuity property, such a mesh can be obtained for bounded polyhedral domains of \( \mathbb{R}^2 \) or \( \mathbb{R}^3 \) with no sharp edges. The extension of the approach to surfaces has not been considered so far and no implementation has been reported for 3-dimensional domains.

### 1.2 Our Contributions

Our work uses the notion of locally uniform anisotropic mesh defined in [Boissonnat et al. 2008], combined with the notion of Delaunay triangulation restricted to a surface. As in the previous approaches, we assume that the anisotropy is prescribed by a metric field that associates to each point \( p \) of the surface, a symmetric positive definite \( 3 \times 3 \) matrix \( M_p \), describing the metric at point \( p \). Given a set of (punctual) sites \( V \) and \( v \in V \), it is easy to compute the 3D Delaunay triangulation \( \text{Del}_3(V) \) for the metric \( M_v \) of \( v \). Indeed, \( \text{Del}_3(V) \) is just the image of an Euclidean Delaunay triangulation under a stretching transformation. For each site \( v \in V \), we consider the star \( T_v \) of \( v \) in \( \text{Del}_3(V) \), and the surface star \( S_v \). The star \( T_v \) is the subcomplex of \( \text{Del}_3(V) \) formed by the tetrahedra that are incident to \( v \), while the surface star \( S_v \) is the subcomplex formed by the triangular facets incident to \( v \) and whose dual Voronoi edges intersect \( S \). The surface star \( S_v \) is thus a collection of triangles incident to \( p \).

In general, there are inconsistencies among the surface stars of the sites \( v \). A facet \( \tau \) appearing in the surface stars of some of its vertices, may not appear in the surface stars of all of them. As a result, the set of surface stars of \( V \) does not form a triangulated surface.

In [Boissonnat et al. 2008; ], we showed that it is possible to refine the set \( V \) of sites until there is no more inconsistencies among the set \( T(\mathcal{V}) \) of 3D stars. This leads to a locally uniform anisotropic 3D mesh in which the star of each vertex is Delaunay with respect to the metric of this vertex. In the present paper, we are interested in generating surface meshes. The algorithm will incrementally compute a set \( V \) of sites on the surface and will maintain the associated surface star set \( S(\mathcal{V}) \). The set \( V \) is refined until there is no inconsistencies among the facets of \( S(\mathcal{V}) \). In contrast to [Boissonnat et al. 2008], we do not consider inconsistencies among the 3D stars. When there is no more inconsistencies among the surface stars, the facets of these surface stars form a locally uniform anisotropic surface mesh that approximates \( S \) and conforms to the specified metric field. This algorithm is provably correct and much more efficient than the 3D anisotropic mesh generator of [Boissonnat et al. 2008].

In addition to conforming to the given metric field, this new method has several notable advantages over previous methods.

- The algorithm relies on the well-established concepts of restricted Delaunay triangulations and Delaunay refinement and comes with theoretical guarantees.
- It can handle complicated geometries and topologies, as well as very anisotropic metric fields.
- It can handle various types of surfaces, e.g. implicit surfaces, polyhedra, isosurfaces in 3D images.
- It is easy to implement and robust since it relies on the usual Delaunay predicates (applied to some stretched spaces).
- The facets of the output mesh have a good aspect ratio as measured with respect to the metric field.

The idea of maintaining independent stars for each vertex of a mesh has been first proposed by Shewchuk [Shewchuk 2005] for maintaining triangulations of moving points. This idea has also been used to build the dual of an anisotropic Voronoi diagram as defined by Labelle and Shewchuk [Schoen 2008]. The idea of stitching stars restricted to a surface can be found in the work on the tangential complex [Boissonnat and Ghosh 2010]. The context is however different from ours since no anisotropy was considered and the main motivation was to reconstruct a \( k \)-manifold embedded in a \( d \)-dimensional space without constructing any global \( d \)-dimensional triangulation, which is not a critical issue in 3 dimensions (see however the discussion in Section 5.3). A major technical difference with our paper is that the stars considered in [Boissonnat and Ghosh 2010] are all contained in a global Euclidean \( d \)-dimensional Delaunay triangulation, which is not the case here. Combining ideas from the present paper and [Boissonnat and Ghosh 2010] would allow to define and generate anisotropic meshes of smooth submanifolds embedded in high dimensional spaces.
2. PRELIMINARIES

2.1 Anisotropic Metric

An anisotropic metric in $\mathbb{R}^3$ is defined by a symmetric positive definite quadratic form represented by a $3 \times 3$ matrix $M$. The distance between two points $a$ and $b$, as measured using a metric $M$, is defined as

$$d_M(a, b) = \sqrt{(a-b)^t M(a-b)}.$$ 

This definition can be extended to $M$-lengths and $M$-areas.

In the following, we often use the same notation, $M$, for a metric and the associated matrix in a given basis. Given the symmetric positive definite matrix $M$, we denote by $F_M$ any matrix such that $\det(F_M) > 0$ and $F_M^t F_M = M$. Note that

$$d_M(a, b) = \sqrt{(a-b)^t F_M^t F_M (a-b) - \|F_M(a-b)\|}$$

where the notation $\| \|$ stands for the Euclidean norm.

Given some metric $M$, an $M$-sphere $C_M(c, r)$, with center $c$ and radius $r$, is defined as the set of points $p$ such that $d_M(c, p) = r$. Likewise, an $M$-ball $B_M(c, r)$, is defined as the set of points $p$ such that $d_M(c, p) \leq r$. Note that an $M$-sphere is an Euclidean ellipsoid, with its axes aligned along the eigenvectors of $M$.

Given a simplex $\tau$ in $\mathbb{R}^3$ and a metric $M$, we define the $M$-circumsphere $C_M(\tau)$ as the smallest $M$-sphere that circumscribes $\tau$. The $M$-circumball $B_M(\tau)$ is the $M$-ball bounded by $C_M(\tau)$ and the $M$-circumradius $r_M(\tau)$ of a simplex $\tau$ is the radius of $C_M(\tau)$.

2.2 Distortion between Metrics

The following definition is due to Labelle and Shewchuk [Labelle and Shewchuk 2003]. Given two metrics $M$ and $N$, and their square-roots $F_M$ and $F_N$, the distortion between $M$ and $N$ is defined as

$$\gamma(M, N) = \max\{\|F_M F_N^{-1}\|, \|F_N F_M^{-1}\|\},$$

where $\| \|$ is the matrix norm operator associated with the Euclidean norm, i.e., for a $3 \times 3$ square matrix $A$, $\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}$. Observe that $\gamma \geq 1$ and $\gamma = 1$ iff $M = N$. A fundamental property of the distortion $\gamma(M, N)$ is to relate distances $d_M$ and $d_N$. Specifically, for any $x$, $y$, we have

$$1/\gamma(M, N) \leq d_N(x, y) \leq d_M(x, y) \leq \gamma(M, N) d_N(x, y).$$

In the rest of the paper, we consider a compact closed surface $S$ of $\mathbb{R}^3$ endowed with a continuous metric field $\{M_p, p \in S\}$ defined over $S$. For convenience, we will simply write $X_c$ instead of $X_M$, when a quantity $X$ is measured using metric $M_\tau$. Hence, we will write for instance $F_p$ for $F_{M_p}$ and $d_p(a, b)$ for $d_{M_p}(a, b)$.

Given two points $p$ and $q$ of $S$, the relative distortion between $p$ and $q$ is defined as $\gamma(p, q) = \gamma(M_p, M_q)$. Note that $\gamma$ is finite since $S$ is compact.

Let $\gamma_0$ be some positive constant called the distortion bound. The distortion radius at $p$, $dr(p, \gamma_0)$ is the upper bound on distances $\ell$ such that for all $q$ and $r$, in $S$:

$$\max(d_p(q, p), d_p(p, r)) \leq \ell \Rightarrow \gamma(q, r) \leq \gamma_0.$$ 

It can be shown that the distortion radius satisfies the following continuity property [Boissonnat et al. 1]:

$$\frac{1}{\gamma(p, q)} (dr(p, \gamma_0) - d_p(p, q)) \leq dr(q, \gamma_0) \leq \gamma(p, q) (dr(p, \gamma_0) + d_p(p, q))$$

2.3 Sizing Field

In addition to controlling the distortion of the metric (which is done by means of the distortion radius), it is important to control the topology of the output mesh. To do so, we adapt the notion of local feature size to the anisotropic setting. The (anisotropic) local feature size at a point $p \in S$, denoted by $h_L(p)$, is defined as the $M_p$-distance from $p$ to the $M_p$-medial axis of $S$. It can be proved that the anisotropic local feature size enjoys a continuity property analogous to (2).

To control both distortion and topology, we use a function on $S$, named the sizing field and defined as $s_l(p, \gamma_0) = \min(h_L(p), dr(p, \gamma_0))$. Plainly, the sizing field $s_l(p, \gamma_0)$ satisfies a continuity property analog to (2). In fact, the user can define the sizing field to be any function provided that it is less than the above function and satisfies a continuity property analog to $s_l$.

3. STARS AND REFINEMENT

3.1 Anisotropic Star Set

Given a metric $M$ and a set of punctual sites $V$, the Delaunay triangulation $D(M, V)$ for metric $M$ is the triangulation of $V$ such that the $M$-circumcircle of each tetrahedron is empty. By empty, we mean that the interior of the $M$-circumcircle contains no points of $V$. By (1), the Delaunay triangulation $D(M, V)$ of a finite set of points $V$ for metric $M$ is simply obtained by computing the Euclidean Delaunay triangulation of the stretched image $F_M(V)$, and then stretching back the result using $F_M^{-1}$.

For each site $v \in V$, we consider the Delaunay triangulation $D(V)$ of $V$ for metric $M_v$. We define the star $T_v$ of site $v$ as the subcomplex of $D(V)$ formed by the tetrahedra that are incident to $v$. The collection of all the stars $T_v$, $v \in V$, is called the star set of $V$. We denote it by $T(V)$.

We now adapt to the anisotropic setting the definition of a restricted Delaunay triangulation. The surface star of $v$, denoted by $S_v$, is the subcomplex of $T(v)$ formed by the facets of $T(v)$ that are incident to $v$ and whose dual Voronoi edge intersects $S$. The collection of the surface stars $S_v$, $v \in V$, is called the surface star set of $V$. We denote it by $S(V)$.

Let $N$ be a metric and $\tau$ be a triangle with its three vertices $p_0, p_1$, and $p_2$ on $S$. We define the $N$-axis of $\tau$ as the set of centers of all the $N$-circumballs of $\tau$. If $p_0, p_1$, and $p_2$ are sufficiently close to each other, and if $\tau$ is well-shaped with respect to metric $N$, the $N$-axis of $\tau$ intersects $S$. We call surface $N$-circumball of $\tau$, and note $B_N(\tau)$, the smallest $N$-circumball of $\tau$ centered on $S$. A facet $\tau \in T_v$ incident to $v$ belongs to $S_v$ iff its surface $M_v$-circumball is empty.

The facet $\tau$ with vertices $p_0, p_1$, and $p_2$ on $S$ has three surface circumballs $B_N(\tau)$ associated respectively to the metrics $M_v$ of each of its vertices. We define the distortion $\gamma(\tau)$ of $\tau$ as the maximal distortion between any pair of points of $S$ which are both inside the same surface circumball $B_N(\tau)$.

3.2 Inconsistencies

Two surface stars $S_v$ and $S_w$ are said to be inconsistent if some facet incident to $v$ and $w$ appears in only one of the two stars $S_v$ and $S_w$. Any facet that appears in the stars of some of its vertices, but not in the stars of all of them, is also said to be inconsistent (see Figure 1).

Our algorithm incrementally inserts new sites in $V$ to resolve inconsistencies among the surface stars.
3.3 Quasi-Cosphericity

Let \( \gamma_0 > 1 \) be a bound on the distortion and \( M \) be a metric. Following [Boissonnat et al. 2008], we now introduce the notion of \((\gamma_0, M)\)-cosphericity and show its link to inconsistent simplices.

Let \( \tau \) be a facet of some surface star and \( p \in V \setminus \tau \). We call \( U = (\tau, p) \) a \((\gamma_0, M)\)-cospherical configuration if there exists two metrics \( N \) and \( N' \) such that

1. \( \gamma(M, N) \leq \gamma_0 \) and \( \gamma(M, N') \leq \gamma_0 \);
2. \( p \) is included in the interior of the surface \( N'\)-circumball \( B_{N'}(\tau) \) and not in the interior of the surface \( N\)-circumball \( B_N(\tau) \).

If \( M \) is clear from the context, we simply say that \( U \) is a \((\gamma_0, M)\)-cospherical configuration and if both \( M \) and \( \gamma_0 \) are understood, we say that \( U \) is a quasi-cospherical configuration.

See Figure 2 for an illustration where \( S \) is the plane.

4. MESHING ALGORITHM

4.1 Algorithm Outline

Let \( S \) be a given surface we want to mesh. The algorithm constructs the set of sites \( V \) by inserting new sites on \( S \) in a greedy way. While there remain inconsistent or bad facets in the surface star set, the algorithm selects one such facet \( \tau \) and kills \( \tau \) by inserting a new site on \( S \) called the refinement point of \( \tau \). Inserting a new point is performed by the following Insert procedure that maintains the star sets \( T(V) \) and \( S(V) \). We say that a tetrahedron \( \sigma \) of a star \( T_w \) conflicts with a point \( p \) if \( p \) belongs to the \( M_w\)-circumball of \( p \).

\( \text{Insert}(p) \)

1. insert \( p \) in each star \( T_v \) that contains a simplex in conflict with \( p \) and update the surface star \( S_v \);
2. create the new stars \( T_p \) and \( S_p \).

The refinement algorithm consists in applying the following rules with a priority order: Rule (i) is applied only if no Rule (j), \( j < i \), can be applied. The algorithm ends when no rule applies anymore. The algorithm relies on procedure Insert to insert a new point in the data structures, and on procedure Pick_valid to select the location of the new site (see the next section).

If \( \tau \) is a facet in some star \( S_v \), we write \( B_v(\tau) \) for the surface \( M_v\)-circumball of \( \tau \), and \( c_v(\tau) \) and \( r_v(\tau) \) for the center and the radius of this ball. The quality of facet \( \tau \) with respect to metric \( M_v \) is characterized by the \( M_v\)-radius-edge ratio, \( \rho_v(\tau) \), which measures the ratio between the \( M_v\)-circumradius of \( \tau \) and the \( M_v\)-length of the edge of \( \tau \) of shortest \( M_v\)-length.

The refinement rules depend on the constants \( \alpha, \rho_0 \) and \( \gamma_0 \).

1. **Sizing field - Distortion**
   - If \( v \in V \) and \( \tau \in S_v \) such that \( r_v(\tau) \geq \alpha sf(c_v(\tau), \gamma_0) \), then Insert\((c_v(\tau))\);
2. **Shape - Radius-edge ratio**
   - If \( v \in V \) and \( \tau \in S_v \) such that \( \rho_v(\tau) > \rho_0 \), then Insert\((\text{Pick_valid}(\tau, M_v))\);
3. **Inconsistency**
   - If \( v \in V \) and \( \tau \in S_v \) such that \( \tau \) is inconsistent, then Insert\((\text{Pick_valid}(\tau, M_v))\),

Assume that the algorithm terminates. Then, all surface stars are consistent and can be stitched together to form a triangulated surface \( T \) with the property that the star of any vertex \( v \) in \( T \) is Delaunay for metric \( M_v \). Hence, \( T \) is a locally uniform anisotropic mesh of surface \( S \). The next step is therefore to establish that the algorithm terminates (Section 4.3).

4.2 Picking Region and Hitting Sets

The simplest idea to kill a bad triangle \( \tau \) in some surface star \( S_v \) is to insert a new site at the center of the surface \( M_v\)-circumball of \( \tau \). However, this simple strategy may lead to endless occurrences of quasi-cosphericities and inconsistencies. We present in this section an alternative strategy, inspired by the strategy introduced by Li
and Teng [Li and Teng 2001; Li 2003] to avoid slivers in isotropic meshes and also used in [Boissonnat et al. 2008]. The basic idea is to relax the choice of the refinement point of a bad facet. Instead of using the center of the surface circumball, the refinement point is picked from a small region around this center, and carefully chosen so as to avoid the formation of new quasi-cosphericities.

Let $\delta < 1$ be a constant called the picking ratio. The $M_v$-picking region of a facet $\tau$ in star $S_v$, denoted by $P_v(\tau)$, is the intersection of the $M_v$-ball $B_{M_v}(c_v(\tau), \delta r_v(\tau))$ with the surface $S$.

In fact, it is not possible, when choosing a refinement point in the picking region $P_v(\tau)$ of a triangle $\tau$ of $S_v$ to avoid the formation of any new quasi-cospherical configurations. The Pick_valid procedure will only avoid the creation of small quasi-cospherical configurations where the meaning of small depends on an additional parameter $\beta \geq 1$ and is made precise below.

Assume we are choosing a new site $p$ in the $M_v$-picking region of some triangle $\tau$ in surface $S_v$. A subset $\sigma$ of three sites of the current set $V$ is said to hit $p$ if the configuration $U = (\sigma, p)$ is both:

1. $(\gamma_0, M)$-cospherical for some metric $M$ such that $\gamma(M_p, M) \leq \gamma_0$, 2. small, meaning that the smallest $M$-circumradius of all the triangles with vertices in $U$ is less than $\beta r_v(\tau)$.

The picking region of triangle $\tau$. Set $\sigma = \{q, r, s\}$ is a hitting set for the picking region $P_v(\tau)$. It defines a forbidden region, shown with red dashes, to be avoided by the refinement point $p$ of $\tau$.

A subset $\sigma$ of three sites of $V$ that hits some point $p$ in the picking region $P_v(\tau)$ is called a hitting set of $P_v(\tau)$. We associate to each hitting set a so-called forbidden region that consists of the points $P_v(\tau)$ hit by $\sigma$. A point of the picking region $P_v(\tau)$ is called a valid refinement point when it does not belong to any forbidden region. See Figure 3.

Note that the definition of a valid refinement point depends on the constants $\delta$ and $\beta$ that respectively define the size of the picking regions and bounds the minimum size of new quasi-cospherical configurations. It also depends on the constant $\gamma_0$ that defines quasi-cosphericity. The next lemma is fundamental for our algorithm.

**Lemma 2** Pick Valid Lemma. For any positive values of the parameters $\alpha \leq 1$, $\beta \geq 1$, $\delta < 1$ and $\rho_0$, it is possible to choose a distortion parameter $\gamma_0$ close enough to 1, so that the valid refinement points form a subregion of positive area inside any picking region considered by the algorithm.

The proof, only sketched here, is given in [Boissonnat et al.]. It uses a volume argument based on the two following facts. 1. The number of hitting sets of a picking region is bounded. 2. For given values of parameters $\alpha$, $\beta$, $\delta$ and $\rho_0$, the volume of the forbidden region associated to a hitting set tends to zero when $\gamma_0$ tends to 1.

To find a valid refinement point in the $M_v$-picking region $P_v(\tau)$ of some bad $k$-simplex $\tau$, the insertion algorithm calls the following Pick_valid procedure that randomly selects a point in the picking region $P_v(\tau)$ until a valid refinement point is found. This procedure depends on the constants $\gamma_0$, $\delta$ and $\beta$, to be fixed later in Section 4.3.

**Pick_valid**($\tau$, $M_v$):

1. Pick randomly a point $p$ in the picking region $P_v(\tau)$
2. If there exists a subset of three sites in $V$ that hits $p$, then discard $p$ and go back to 1.

Lemma 2 ensures that the picking process will succeed with a positive probability at each trial. Observe also that Pick_valid takes care of small configurations only. Hence, checking if a point of some picking region is valid requires only to consider sites that are close to the picking region. Furthermore, the number of triplets we need to consider as possible hitting sets is bounded by a constant (see the proof of Lemma 2 in [Boissonnat et al.]).

### 4.3 Termination of the Algorithm and Quality of the Mesh

We now prove that, under some conditions on the parameters $\rho_0$, $\gamma_0$, $\delta$ and $\beta$, the number of inserted vertices is bounded, which implies that the algorithm terminates. Let us define the intersite distance between two sites $p$ and $q$ as $d(p, q) = \min(d_v(p, q), d_h(p, q))$, and the insertion radius $r(p)$ of a site $p$ as the minimum intersite distance between $p$ and the current set of sites, i.e., the set of sites that have been inserted before $p$. Recall that $\Gamma = \sup_{x, y \in S} \gamma(x, y)$ is finite since $S$ is compact. The proofs of the following lemmas, available in [Boissonnat et al.], are only sketched here.

**Lemma 3** Insertion Radius Lemma. If the two conditions

$$\frac{(1 - \delta)\rho_0}{1 - \gamma_0} \geq 2 \quad \text{and} \quad \frac{(1 - \delta)\beta}{\Gamma(1 + \delta)} \geq 2$$

hold, then there exists a constant $C$ such that the insertion radius $r(p)$ of any vertex $p$ of the mesh is at least $Cs(p)$.

The proof uses the fact that the new point is inserted close to the center of the surface circumball of a facet in a surface star, and therefore relatively far from the current set of sites. The precise proof is a case analysis that studies the three rules of the algorithm. From this lemma, we can bound from below the intersite distance between any two points $p$ and $q$ in the final mesh. We consider whether $p$ or $q$ has been inserted last. Lemma 3 and the continuity property (2) of $s$ then allow to prove the following lemma.

**Lemma 4** Smallest Intersite Distance. If the conditions of Lemma 3 are satisfied, the smallest intersite distance between a site $p$ and all the other vertices of the final mesh is at
least: 
\[ d(p, q) \geq \frac{C}{(1 + C)\gamma} \text{sf}(p). \]

It follows from the lemma that the \( M_p \)-balls \( B_p(l(p)) \), where \( l(p) = \frac{C}{2\gamma(p) + C} \text{sf}(p) \), have disjoint interiors. A simple volume calculation then provides an upper bound on the number of vertices of the final mesh.

**Lemma 5**: Number of Inserted Sites. If the conditions of Lemma 3 are satisfied, the number \( n \) of sites inserted by the algorithm is at most

\[
n \leq \frac{\delta}{\pi} \left(1 + \frac{2(1 + C)}{C}\right)^2 \int_S \text{sf}(p)^2 \, ds\]

When the algorithm terminates, the surface stars are consistent and can be stitched so as to make a triangulated surface mesh \( S \). Each triangular facet of \( S \) is Delaunay for the metric of any of its three vertices, conforms to the sizing field \( \text{sf} \) (Rule 1) and has a radius-edge ratio smaller than \( \rho_0 \) (Rule 2) when measured using the metric of any of its vertices. The radius-edge ratio being a fair quality measure for triangles [Shewchuk 2002], the last property ensures that the facets of the output triangulated surface are well-shaped with respect to the local anisotropy.

In addition, we can ensure that the topology of \( S \) is the same as the topology of \( S \). This can be done by choosing a sizing field that is sufficiently small with respect to the local feature size of the surface. This condition, together with the bound on the radius-edge ratio of the facets, ensures that \( S \) is homeomorphic (and even ambient isotopic) to \( S \). See for example [Amenta et al. 2002; Boissonnat and Oudot 2005].

### 5. IMPLEMENTATION

The algorithm can handle different kinds of surfaces provided that we are given an oracle able to:

1. compute a few initial points on the surface,
2. detect and compute the intersections between a given line segment and the surface.

A small set of initial points is enough. We create the initial stars and put the bad facets of the surface stars in a global queue containing all the facets to be refined. The refinement process can then start. Our implementation uses the CGAL library [CGAL].

#### 5.1 The algorithm parameters

Our anisotropic surface mesh generation algorithm depends on five parameters: \( \alpha, \beta, \delta, \rho_0 \), and \( \gamma_0 \). A priori, the parameters values should be chosen to satisfy the conditions of Lemma 3 and \( \gamma_0 \) should be set close enough to 1 to ensure the existence of valid refinement points in any picking region ( Lemma 2). Note however that the conditions in Lemma 3 are not tight. They are in fact widely influenced by our will to keep the proof simple. In practice, the meshing algorithm runs smoothly and yields good meshes for a larger range of the parameter values.

Parameter \( \alpha \) should be chosen \( \leq 1 \). This parameter is needed to prove the picking lemma. In practice, this parameter is not critical and has been set to 1 in all our experiments.

Parameters \( \beta \) and \( \delta \), which define the picking region and the validity of a refinement point, have little influence on the quality of the final mesh. Those parameters have to be set in such a way that procedure \( \text{PickValid} \) finds a valid refinement point in a reasonable amount of time. In any case, we must have \( \beta > 1 \) and \( \delta < 1 \). In all our experiments, we use \( \beta = 2.5 \) and \( \delta = 0.3 \).

Parameter \( \rho_0 \), on one hand, should be as small as possible to ensure that the simplices of the final mesh are well-shaped according to the metric field. On the other hand, \( \rho_0 \) must not be less than 1 to ensure that the refinement algorithm terminates. In all our experiments, we use \( \rho_0 = 3 \).

The size of the simplices in the final mesh mostly depends on the parameter \( \gamma_0 \) and on the sizing field. As defined above, the sizing field takes into account the variation of the metric (distortion radius) and the geometry of the surface (local feature size). Estimating the local feature size is a notoriously difficult problem, for which existing solutions can be used [Boissonnat and Oudot 2005; Antani et al.]. In this paper, the emphasis is put on the metric variation and, in our experiments, we assume a local feature size that is uniformly bounded from below by some parameter \( \gamma_0 \) that we use to specify the wished size of the simplices in the mesh. To take into account the metric variation, we replace the distortion radius by a direct and simpler control on the distortion of the simplices. Specifically, Rule 1 is replaced in practice by the following rule:

1. **Sizing field - Distortion**: If \( \gamma \in V \) and \( \tau \in S_\gamma \), such that either \( r_v(\tau) \geq \gamma_0 \) or \( \gamma(\tau) \geq \gamma_0 \), then \( \text{Insert}(\gamma_0(\tau)) \).

Parameter \( \gamma_0 \) has to be greater than 1 and close enough to 1. The influence of the value of parameter \( \gamma_0 \) is complex. On one hand, choosing for \( \gamma_0 \) a value very close to one triggers many application of the Sizing field - Distorsion rule and leaves only few inconsistencies to be removed by the Inconsistency rule. Such a choice also keeps the area of the forbidden regions small, which eases picking valid refinement points. On the other hand, such a choice may lead to an overly dense mesh. Raising the value of \( \gamma_0 \) results in a larger number of inconsistencies to be solved by the Inconsistency rule, an increasing number of calls to procedure \( \text{PickValid} \) and therefore a slow down of the meshing process. In all our experiments, \( \gamma_0 \) is chosen between 1.3 and 1.8. Also, for efficiency reasons, we approximate \( \gamma(\tau) \) by the maximum distortion between two vertices of \( \tau \). The influence of parameter \( \gamma_0 \) on the resulting mesh density and algorithm efficiency is shown on Figure 7.

In summary, our implementation depends on parameters \( \beta, \delta, \rho_0, \gamma_0 \), and \( \gamma_0 \). In all our experiments, \( \beta, \delta, \rho_0 \) have been set to fixed values.

#### 5.2 Computing the Metric Field

The algorithm requires to compute the metric field at any refinement point on \( S \). It can either be prescribed by the user or approximate the curvature tensor of the surface at the refinement point. We detail below several cases to illustrate the versatility of our approach.

##### 5.2.1 Three-Dimensional Scalar Fields

Three-dimensional scalar fields are common in physical or mathematical applications. If \( E \) denotes such a scalar field defined in \( \mathbb{R}^3 \) and \( S \) a surface, we may want to mesh \( S \) so as to provide the best approximation of \( E \) on \( S \). The triangles should then be elongated orthogonally to the gradient \( \nabla E \). We define the metric field as follows.

\[
F_M = U \cdot \text{diag}(1/(1 + \varphi(\|\nabla E\|)), 1, 1) \cdot U^t
\]

where \( U = [\frac{\nabla E}{\|\nabla E\|}, U_2, U_1] \) and \( U_2 \) being two arbitrary unit vectors that form an orthogonal frame with \( \nabla E \). If the gradient is
zero, the metric field is isotropic. Parameter $\omega$ controls how the facets are stretched with respect to the norm of the gradient.

Figure 4 presents an anisotropic Delaunay mesh of a sphere endowed with a sine-shaped shock scalar field. The triangles are elongated orthogonally to the gradient direction and the density depends both on the norm of the gradient and on the changing rate of the gradient.

5.2.2 Curvature Tensor on an Implicit Surface. Let $S$ be an implicit surface defined by $f(x, y, z) = 0$. At each point $p$ of $S$, we denote by $\nu$ the normal vector $-\nabla f / \| \nabla f \|$ and by $H$ the Hessian of $f$. The principal curvatures $\kappa_{\text{max}}, \kappa_{\text{min}}$ are the non-zero eigenvalues of the matrix

$$C = P \cdot H' \cdot P$$

where $H' = H / \| \nabla f \|$, $P = I_3 - N \cdot N^t$, see e.g. [Hughes 2003]. Let $U_{\text{max}}, U_{\text{min}}$ and $N$ be the normalized eigenvectors of $C$, and let $U = [U_{\text{max}}, U_{\text{min}}, N]$. The metric at point $p$ is defined by $M_p = P_p^t F_p$, where:

$$F_p = U \cdot \Delta \cdot U^t$$

with $\Delta = \text{diag}(\kappa_{\text{max}}, \kappa_{\text{min}}, \epsilon)$, $\kappa_{\text{max}} = \max \{ \epsilon, \| C_{\text{max}} \| \}$, $\kappa_{\text{min}} = \max \{ \epsilon, \| C_{\text{min}} \| \}$.

Here, $\epsilon$ is a small positive constant that prevents the eigenvalues of the metric to vanish when $\kappa_{\text{max}}$ and/or $\kappa_{\text{min}}$ vanish, e.g., at planar or parabolic points. $\epsilon$ is a constant global to be discussed in the next subsection, and $\Delta = \text{diag}(\kappa_{\text{max}}, \kappa_{\text{min}}, \epsilon)$ stands for a diagonal matrix whose entries are $\{ \kappa_{\text{max}}, \kappa_{\text{min}}, \epsilon \}$.

Figure 5 shows an anisotropic Delaunay mesh of the “chair” implicit surface.

5.2.3 Curvature Tensor on Polyhedral Surfaces. To remesh a polyhedral surface $S$ that may be an approximation of a smooth surface $S'$, we can estimate the curvature tensor of $S'$ from $S$. This is done in two steps.

1. Curvature estimation at the vertices of $S$
   We first estimate the curvature tensor at each vertex $v \in S$ as follows. We compute the Euclidean $k$-nearest vertices of $v$, and then apply the jet-fitting algorithm of [Cazals and Pouget 2005] to estimate the two principal curvatures $\kappa_{\text{max}}$ and $\kappa_{\text{min}}$, the corresponding unit vectors $U_{\text{max}}$ and $U_{\text{min}}$, and the normal vector $N$ at $v$. From the estimated curvatures, we compute $\kappa_{\text{max}}$ and $\kappa_{\text{min}}$ as above. We choose $\epsilon$ as explained in subsection 5.4 and store at each vertex $v$ a resulting estimated metric tensor $M(v) = U \Delta^2 U^t$, with $U = [U_{\text{max}}, U_{\text{min}}, N]$ and $\Delta = \text{diag}(\kappa_{\text{max}}, \kappa_{\text{min}}, \epsilon)$.

2. Blending metric tensors
   To estimate the metric tensor at any prescribed query point on $S$, we smoothly blend the metric tensors computed in the previous step using the following formula:

$$M_p = \sum_{v \in S} \omega_p(v) M(v), \quad \omega_p(v) = \frac{\mu_p(v)}{\sum_{v \in S} \mu_p(v)}$$

where $r_B$ is a parameter that controls the locality of the blending. By default, $r_B$ can be computed by averaging the distance to the $k$-nearest points of each vertex in $S$.

$M_p$ is a positive combination of symmetric tensors and therefore has no negative eigenvalues. This property can be proved by contradiction. Assume that $M_p$ has some negative eigenvalues. Then there exists a vector $x$ that satisfies $x^t M_p x = \sum_{v \in S} \omega_p(v) x^t M(v) x < 0$. Since all $M(v)$ are symmetric tensors, $x^t M(v) x \geq 0$ and, since according to the definition, $\omega_p(v) \geq 0$, we have $x^t M_p x \geq 0$, which provides a contradiction.

Figure 6 shows an anisotropic Delaunay mesh of a polyhedral surface conforming to the estimated curvature metric field.

5.3 Computing Surface Stars
Our algorithm needs to maintain the surface star $S_v$ of each vertex $v$ in the mesh. As described above, $S_v$ can be extracted from the star $T_v$ of $v$ in Del$_{\text{vol}}(V)$. Yet, maintaining stars $T_v$ is not a fully satisfactory solution. Indeed, since the vertices of the mesh all lie on the surface, most of the tetrahedra in stars $T_v$ spread through space and their diameters do not decrease significantly when the sample density increases. As a consequence, the algorithm is not local and the computing time is not optimized. To remedy this problem, we add a few vertices close to the medial axis of the surface and maintain for each vertex $v$ a triangulation $T_v'$ that does not necessarily include the star $T_v$, but nevertheless has the same restricted star $S_v$. 

Fig. 4. A unit sphere endowed with a metric field derived from the sine-shaped shock function $E(x, y, z) = \tanh(1 / 2 (2x - \sin(5y))) + x^3 + xy^2$, $\lambda = 0.6$, $\rho_0 = 3.0$, $r_0 = 0.1$, $\gamma_0 = 1.4$, $\beta = 2.5$ and $\delta = 0.3$. The mesh has 18323 vertices and 36642 facets.
as $T_v$. The vertices close to the medial axis are computed from the initial points following Amenta and Bern's poles method [Amenta and Bern 1999]. The triangulation $T'_v$ is defined by the following insertion rule: a site $p$ is inserted in the triangulation $T'_v$ of a site $v$ only if $p$ belongs to the so-called conflict zone of $T'_v$. The conflict zone of a surface star $S_v$, denoted by $Z_v$, is the union of the surface $M_r$-circumballs $B_r(x)$ for all the facets $r$ of $S_v$. The correctness of this variant of the algorithm follows from the fact that the conflict zone of star $S_v$ can only shrink when inserting a new site $p$, and from the fact that the surface star $S_v$ has to be updated iff $p$ belongs to $Z_v$.

When the sample density increases, the mesh facets become small and have small surface circumballs. At each insertion of a new site $p$, we use the conflict zones as filters to quickly report the vertices $v$ whose small triangulations $T'_v$ have to be updated and the vertices $w$ that have to be inserted in the triangulation $T'_w$ created for such site.

To each conflict zone $Z_v$, we associate its (Euclidean) axis-parallel bounding box that we simply call the bounding box of $v$ and denote by $B_v$. We maintain in a data structure $Q$ the bounding boxes of the current set of vertices $V$ and we maintain $V$ in a kd-tree $Kd(V)$. When inserting a new site $p$, we first query the data structure $Q$ and report all the boxes that contain $p$. If $p$ belongs to some box $B_w$, we further check whether $p$ belongs to the conflict zone $Z_w$ and, in the affirmative, we insert $p$ in the triangulations $T'_w$. We then build the triangulation $T'_w$ of the new site. $T'_w$ is initialized as the star of $p$ in the 3D Delaunay triangulation of $p$ and of all the vertices $v$ whose triangulations $T'_v$ have been updated by inserting $p$. Then, we compute a bounding box for the current conflict zone $Z_p$, and query the kd-tree $Kd(V)$ to compute all the vertices in this box. Each vertex in this box is inserted in $T'_p$ if it belongs to the current conflict zone $Z_p$.

5.4 Sliver Prevention

When the sampling is dense enough, quasi-cospherical configurations are tetrahedra with well-shaped facets (because of Rule 2) that have small volumes and are almost tangent to the surface. We call such tetrahedra slivers. We can reduce the number of slivers by enlarging the eigenvalue of the metric in the direction normal to the surface (eigenvalue $e_3$ defined in the preceding subsection). Indeed, the volume of those tetrahedra in the new metric will be increased, hence reducing the number of slivers and inconsistencies. Observe that this modification of the metric does not significantly affect the shape of the facets of the mesh since those facets are almost tangent...
to the surface. Experimental evidence indicates that \(e_{\text{m}}\) should be of the same order of magnitude as the global maximum of \(c_{\text{max}}\).

6. RESULTS

6.1 Performance and Complexity

Our experiments reveal that our algorithm is roughly linear behavior with respect to the number of vertices of the final mesh. As is well-known, for most point sets and insertion orders, inserting a new point in a Delaunay triangulation takes a constant time. Moreover, our use of the conflict zones (Section 5.3) ensures that a constant number of stars are visited when inserting a new vertex and a constant number of vertices are considered when creating the star of the new vertex.

Figure 7 shows computation times for meshing a torus defined by an implicit function. All meshes are generated with \(r_{\text{star}}\) of the new vertex.

![Fig. 7. Computing time to mesh an implicit torus with an implicit function. All meshes are generated with \(r_{\text{star}}\) of the new vertex.](image)

6.2 Mesh Quality

Our meshing algorithm provides surface meshes of guaranteed quality since the algorithm controls the radius-edge ratio of each facet, as measured by the metric of any facet vertex. Figure 8 shows the histogram of the squared radius-edge ratios of the facets. For each facet, three radius-edge ratios were computed, one for each of the metrics of the vertices of the facet.

The quality of the meshes produced by our algorithm can still be improved by locally optimizing the positions of the vertices. This is done by adapting the optimization method of Alliez et al. [Alliez et al. 2005]. Each vertex \(v\) is moved to the barycenter of the circumcenters of the facets in star \(S_{v}\), weighted by their respective areas. In our anisotropic setting, the circumcenters are \(M_{v}\)-circumcenters, the weights are the \(M_{v}\)-areas of the facets of \(S_{v}\), and the barycenter \(b_{v}\) is a \(M_{v}\)-barycenter. In order to preserve the locally uniform property of the anisotropic mesh, the motion should not introduce inconsistent, badly-shaped nor over-distortionized facet. Hence, instead of directly moving \(v\) to \(b_{v}\), we test new positions on the line segment \((v, b_{v})\) and check that the planned motion does not trigger the application of one of the Rules (1)-(3). We first test \(v' = b_{v}\). If \(v'\) is a valid refinement point, we move \(v\) to \(v'\), otherwise, we set \(v' = (v + v')/2\) while \(d_{\text{e}}(v, v') < h\). Each star is optimized in turn. In our implementation, the stars are sorted by decreasing values of \(d_{\text{e}}(v, b_{v})\) and maintained in a priority queue. The first element is optimized first. The algorithm terminates when no more star can be optimized. Figure 8 shows that the distribution of radius-edge ratios is improved after the optimization step. The additional computing time required for this optimization is negligible.

6.3 Accuracy-Size Trade-off

Using an anisotropic metric field derived from the curvature tensor leads to surface meshes that enhance the trade-off between the accuracy of the approximation and the size of the mesh.

To measure the accuracy of a mesh, we use the mean error which we compute by uniformly sampling the mesh and averaging the Euclidean distance between each sample point on the mesh and its projection on the surface.

Figure 9 provides a comparison between isotropic meshes and curvature adapted anisotropic meshes of tori. Each torus has a small radius \(r = 1\), while the big radius \(R\) takes different values. In both the isotropic and the anisotropic cases, the meshes have been obtained by a refinement process where the size criterion \((r_{\text{0}})\) has been replaced by a bound on the quality of the approximation over...
a facet. Hence, both meshes are guaranteed to approximate the surface up to the same error bound. As can be seen, the number of vertices of the anisotropic mesh can be much smaller than the number of vertices of the corresponding isotropic mesh. The figures also enlight the fact that the number of vertices of the anisotropic mesh decreases with the $r/R$-ratio, while the isotropic mesh is barely affected.

7. CONCLUSION

We have introduced a novel approach to generate anisotropic meshes of surfaces. Given a surface $S$ endowed with a metric tensor that varies over $S$, our algorithm produces a mesh that approximates $S$ and whose elements conform to the metric field. The resulting mesh provides a better approximation of the surface (if one takes as metric tensor the curvature tensor of $S$) or a better approximation of a function $f$ defined over $S$ (if we take as metric tensor the hessian of $f$). Our algorithm is simple and comes with theoretical guarantees without relying on heuristics. As demonstrated by our experiments, it can handle complex shapes and metric fields. The algorithm is generic in the sense that it can handle various types of surfaces such as implicit surfaces or triangulated surfaces. Although we have not presented such results, it could be easily extended to mesh isosurfaces in 3D images.

As future work, we intend to consider surfaces with sharp features following the approach of [Dey and Levine 2009]. We also intend to extend our method to evolving surfaces [Li and Teng 2001], and to 3D domains bounded by surfaces and equipped with a 3D metric field, combining the results of this paper with [Boissonnat et al. 2008].

REFERENCES


ANTANI, L., DELAGE, C., AND ALLIEZ, P. Mesh sizing with additively weighted voronoi diagrams.


Fig. 10. Example of an anisotropic mesh of a torus. $R = 50$, $r = 1$, $\text{approx} = 0.001$


