Anisotropic Geodesics for Perceptual Grouping and Domain Meshing [⋆]

Sébastien Bougleux, Gabriel Peyré and Laurent Cohen

Université Paris-Dauphine, CEREMADE, 75775 Paris Cedex 16, France
{bougleux,peyre,cohen}@ceremade.dauphine.fr

Abstract. This paper shows how Voronoi diagrams and their dual Delaunay complexes, defined with geodesic distances over 2D Reimannian manifolds, can be used to solve two important problems encountered in computer vision and graphics. The first problem studied is perceptual grouping which is a curve reconstruction problem where one should complete in a meaningful way a sparse set of noisy curves. From this latter curves, our grouping algorithm first designs an anisotropic tensor field that corresponds to a Reimannian metric. Then, according to this metric, the Delaunay graph is constructed and pruned in order to correctly link together salient features. The second problem studied is planar domain meshing, where one should build a good quality triangulation of a given domain. Our meshing algorithm is a geodesic Delaunay refinement method that exploits an anisotropic tensor field in order to locally impose the orientation and aspect ratio of the created triangles.

1 Introduction and Related Concepts

Various important problems in computer vision and computer graphics require the integration of a local anisotropy over a complicated planar domain. This local anisotropy is dictated by directional features such as curves or textures that should be exploited to perform sampling, segmentation, grouping or meshing tasks. In this paper, we focus on two representative problems in these fields: perceptual grouping of salient features and meshing of planar domains.

The proposed approach encodes the local anisotropy within a tensor field that corresponds to a Riemannian metric. This local information is integrated into global constraints thanks to the geodesic distance over this Riemannian domain. Based on this geodesic distance, one can define the Riemannian Voronoi diagram of a point set [1] and its dual Delaunay complex to perform grouping and meshing tasks. Such geometrical structures are not easy to compute, and several approximate solutions have been proposed (see for instance [2], [3]). Here, we adopt an approach based on the Fast Marching algorithm to compute geodesic distances and Voronoi diagrams.

This section reviews basis facts about Riemannian manifolds and explains how geodesic distances can be computed efficiently with Fast Marching methods.

[⋆] This work was partially supported by ANR Grant SURF-NT05-2_45825.

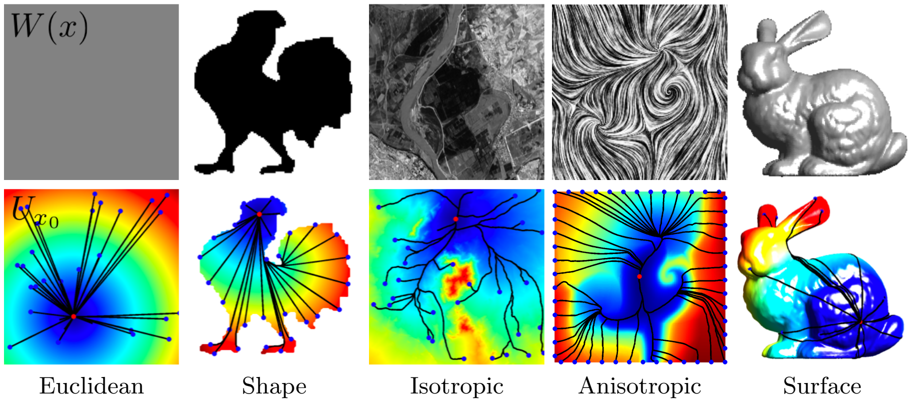


Fig. 1. Examples of Riemannian metrics (top row) and geodesic distances and curves (bottom row). The colormap indicates the geodesic distance to the starting point. From left to right: Euclidean ($H(x) = \mathrm{Id}_2$) in $\mathbb{R}^2$ and then restricted to a planar domain, isotropic ($H(x) = W(x)\mathrm{Id}_2$ with W computed from the image), Riemannian manifold ($H(x)$ is the structure tensor of the image, see (8)) and 3D surface ($H(x)$ corresponds to the first fundamental form).

Riemannian metric, geodesic distances and shortest paths. This paper considers 2D Riemannian manifolds that are defined over a compact planar domain $\Omega \subset \mathbb{R}^2$. The boundary $\partial\Omega$ of Ω is assumed to be a set of closed smooth curves. At each point $x \in \Omega$, one has a tensor $H(x) \in \mathbb{R}^{2\times 2}$ which is a positive symmetric matrix. This tensor field defines a local metric that allows to measure the length of a piecewise smooth curve $\gamma : [0,1] \to \Omega$ as follows

$$L(\gamma) \overset{\text{def.}}{=} \int_0^1 \sqrt{\gamma'(t)^{\mathrm{T}} H(\gamma(t))\gamma'(t)}\mathrm{d}t.$$

In image processing and computer vision, the manifold is often the image domain $\Omega = [0,1]^2$ equipped with a metric H derived from the image to process. In computer graphics and numerical analysis, one often deals with complicated planar domains with holes and corners. Figure 1 shows several Riemannian metrics. An important issue is thus to design, for a specific application, a tensor field H that encodes the important information about the problem to solve.

At each location $x \in \Omega$, the Riemannian tensor can be diagonalized as

$$H(x) = \lambda_1(x)e_1(x)e_1(x)^{\mathrm{T}} + \lambda_2(x)e_2(x)e_2(x)^{\mathrm{T}}, \tag{1}$$

where $\lambda_1 \geqslant \lambda_2 > 0$ are the eigenvalue fields, and e_1, e_2 are the corresponding orthogonal (un-oriented) eigenvector fields. A curve γ passing at location $\gamma(t) = x$ with speed $\gamma'(t)$ has a shorter local length if $\gamma'(t)$ is collinear to $e_1(x)$ rather than to another direction. Hence shortest paths tend to be tangent to the field e_1.

The anisotropy of a metric $H(x)$ is defined as

$$\alpha(x) \overset{\text{def.}}{=} \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} = \frac{\sqrt{(a-b)^2 + 4c^2}}{a+b} \in [0,1], \quad \text{for } H(x) = \begin{pmatrix} a & c \\ c & b \end{pmatrix}. \tag{2}$$

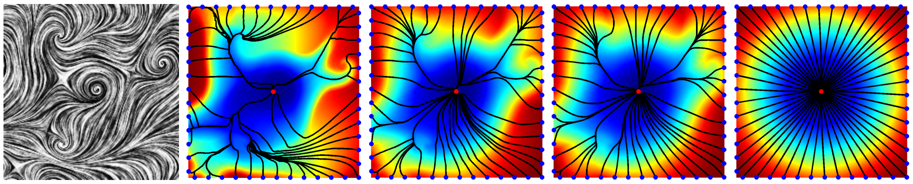


Fig. 2. Distance maps and geodesics for a tensor field with a decreasing anisotropy α. The tensor field is computed from the structure tensor of f as defined in (8).

A metric with $\alpha(x)$ close to 1 is highly directional near x, whereas a metric with $\alpha(x)$ close to 0 is locally isotropic near x.

The geodesic distance between two points of Ω is defined as

$$d(x,y) \overset{\text{def.}}{=} \min_{\gamma \in \mathcal{P}(x,y)} L(\gamma), \quad \forall\,(x,y) \in \Omega^2, \tag{3}$$

where $\mathcal{P}(x,y) \overset{\text{def.}}{=} \{\gamma \setminus \gamma(0){=}x \wedge \gamma(1){=}y\}$ denotes the set of piecewise smooth curves joining x and y. The distance d turns the domain Ω into a metric space.

A shortest path γ between two points $(x,y) \in \Omega^2$, according to the Riemannian metric, is called a geodesic. It satisfies $L(\gamma) = d(x,y)$.

Distance map. In order to perform the numerical computation of geodesic curves and distances, we fix a set of starting points $\mathcal{S} = (x_k)_k \subset \Omega$ and we define the distance map to this set

$$U_{\mathcal{S}}(x) \overset{\text{def.}}{=} \min_k d(x, x_k), \quad \forall\, x \in \Omega.$$

An important theoretical result is that if $x \mapsto H(x)$ is continuous, $U_{\mathcal{S}}$ is the unique viscosity solution of the following Hamilton-Jacobi non-linear partial differential equation (PDE)

$$\|\nabla_x U_{\mathcal{S}}\|_{H(x)^{-1}} = 1, \quad \text{with} \;\; U_{\mathcal{S}}(x_k) = 0, \; \forall k, \tag{4}$$

where $\|v\|_A = \sqrt{v^T A v}$. This classical result (see for instance [4]) allows to replace the optimization problem (3) that defines geodesic distances by a PDE.

Once the distance map $U_{\mathcal{S}}$ has been computed by solving (4), one can extract a geodesic γ between a point x and its closest point $x_k \in \mathcal{S}$ by the following gradient descent

$$\gamma'(t) = -\frac{H(\gamma(t))^{-1} \nabla_{\gamma(t)} U_{\mathcal{S}}}{\|H(\gamma(t))^{-1} \nabla_{\gamma(t)} U_{\mathcal{S}}\|}, \quad \text{with} \;\; \gamma(0) = x. \tag{5}$$

The geodesic curve γ extracted using this gradient descent is parametrized with unit speed since $\|\gamma'\| = 1$. In particular, if $\mathcal{S} = \{x_1\}$ is restricted to one point, then one can compute the geodesic curve between x_1 and any point in Ω.

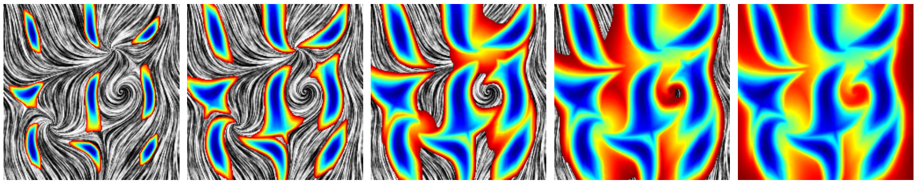


Fig. 3. Anisotropic front propagation from 9 starting points. The colormap indicates the values of the distance map at a given iteration of the FM algorithm. The metric is computed using the structure tensor (see (8)), of the background image.

Figure 2 shows examples of geodesic curves computed from a single starting point $\mathcal{S} = \{x_1\}$ located at the center of the domain $\Omega = [0,1]^2$, and a set of points on the boundary of Ω. The geodesics are computed for a metric H, derived from the image $f : \Omega \rightarrow \mathbb{R}$, and whose anisotropy α (defined in (2)) is decreasing, thus making the Riemannian space progressively closer to the Euclidean space.

For the particular case of an isotropic metric $H(x) = W(x)\mathrm{Id}_x$, the geodesic distance and the shortest path satisfy

$$\|\nabla_x U_{\mathcal{S}}\| = W(x) \qquad \text{and} \qquad \gamma'(t) = -\frac{\nabla_x U_{\mathcal{S}}}{\|\nabla_x U_{\mathcal{S}}\|}. \tag{6}$$

This corresponds to the Eikonal equation, that has been used to compute minimal paths weighted by W [5].

Numerical computations of geodesic distances. In order to make all the previous definitions effective in practical situations, one needs a fast algorithm to compute the geodesic distance map $U_{\mathcal{S}}$. The Fast Marching (FM) algorithm, introduced by Sethian [6], is a numerical procedure to efficiently solve in $O(n\log(n))$ operations the discretization of (6) in the isotropic case. Several extensions of the FM algorithm have been proposed in order to solve (4) for a generic metric, see for instance Kimmel and Sethian [7] for triangulated meshes and Spira and Kimel [8], Bronstein et al. [9] for parametric manifolds.

In our numerical experimentations, we use the $O(n\log(n))$ FM algorithm developed by Prados et al. [10]. As any FM method, it computes the distance map $U_{\mathcal{S}}$ by progressively propagating a front, starting from the points in $\mathcal{S}$. Figure 3 shows an example of FM computation with an anisotropic metric. The front propagates faster along the direction of the texture. This is due to the alignment of the principal direction field e_1 (of the metric) with the textures.

2 Riemannian Voronoi Diagrams and Dual Complexes

This section shows how several concepts from computational geometry can be extended to the setting of Riemannian metrics. Given a 2D Riemannian manifold as defined in Section 1, and a set of points $\mathcal{S} = \{x_i\}_{i=1}^m \subset \Omega$, one can define combinatorial and geometrical structures that organize the domain Ω and the set $\mathcal{S}$ with respect to the geometry of the manifold.

Voronoi diagrams. Riemannian Voronoi diagrams [1] considered in this paper have the same definition as the Euclidean Voronoi diagram and its extensions. Given a metric and a set of points $\mathcal{S}$, the partition $\mathcal{V}_\mathcal{S}$ of the domain Ω in Voronoi cells $\mathcal{C}_i$ is given by

$$\mathcal{V}_\mathcal{S} \overset{\text{def.}}{=} \bigcup_{x_i \in \mathcal{S}} \mathcal{C}_i = \Omega \quad \text{with} \;\; \mathcal{C}_i \overset{\text{def.}}{=} \{x \in \Omega \setminus \forall\, j \neq i,\, d(x_i, x) \leqslant d(x_j, x)\}\,. \tag{7}$$

Using the anisotropic geodesic distance (3) for d, $\mathcal{V}_\mathcal{S}$ is a Riemannian Voronoi diagram of the set S, restricted to the domain Ω. This anisotropic geodesic Voronoi diagram shares several properties with the Euclidean Voronoi diagram. In particular, anisotropic geodesic Voronoi cells have the advantage to be connected. This is not necessarily the case for the so-called anisotropic Voronoi diagram [3], defined with a simplified anisotropic distance $\tilde{d}(x,y) = ((x-y)^\mathrm{T} H(x)(x-y))^{1/2}$ that is not based on shortest paths.

Two Voronoi cells $\mathcal{C}_i$ and $\mathcal{C}_j$ are adjacent if $\mathcal{C}_i \cap \mathcal{C}_j \neq \emptyset$. This intersection defines a Voronoi edge that is the set of points of Ω located at equal distance from the points x_i and x_j of $\mathcal{S}$. With the anisotropic geodesic distance (3), such an edge is an open or closed curve. A Voronoi vertex is the intersection point of at least three adjacent Voronoi cells. When the set $\mathcal{S}$ is in general position, a Voronoi vertex is the intersection point of exactly three Voronoi cells

$$\mathcal{V}_{i,j,k} \overset{\text{def.}}{=} \mathcal{C}_i \cap \mathcal{C}_j \cap \mathcal{C}_k.$$

In the sequel, the set $\mathcal{S}$ is assumed to be in general position, and the set of Voronoi vertices is noted $\Sigma_\mathcal{S}$.

Delaunay complex. The Delaunay complex $\mathcal{D}_\mathcal{S}$ of the set of points $\mathcal{S}$ is the simplicial complex dual to the anisotropic geodesic Voronoi diagram. It is the set of triangles, edges and vertices that can be defined as follows:

- the set of vertices of $\mathcal{D}_\mathcal{S}$ is the set of points $\mathcal{S}$,
- two points of $\mathcal{S}$ are connected by an edge of $\mathcal{D}_\mathcal{S}$ if their respective Voronoi cells are adjacent in $\mathcal{V}_\mathcal{S}$

$$(x_i, x_j) \in \mathcal{D}_\mathcal{S} \iff \mathcal{C}_i \cap \mathcal{C}_j \neq \emptyset.$$

- three points of $\mathcal{S}$ form a triangle of $\mathcal{D}_\mathcal{S}$ if their respective Voronoi cells have a non-empty intersection, that is a Voronoi vertex

$$(x_i, x_j, x_k) \in \mathcal{D}_\mathcal{S} \iff \mathcal{C}_i \cap \mathcal{C}_j \cap \mathcal{C}_k \neq \emptyset.$$

To each Delaunay edge (x_i, x_j) corresponds a double point $w_{i,j}$, defined as the closest point to x_i and x_j, that is on the Voronoi edge $\mathcal{C}_i \cap \mathcal{C}_j$

$$w_{i,j} \overset{\text{def.}}{=} \underset{x \in \mathcal{C}_i \cap \mathcal{C}_j}{\operatorname{argmin}}\, d(x, x_i) \overset{(3)}{=} \underset{x \in \mathcal{C}_i \cap \mathcal{C}_j}{\operatorname{argmin}}\, d(x, x_j).$$

Contrary to the Euclidean case, the Delaunay complex $\mathcal{D}_\mathcal{S}$ defined above does not always define a valid planar triangulation. Some points of $\mathcal{S}$ can be isolated,

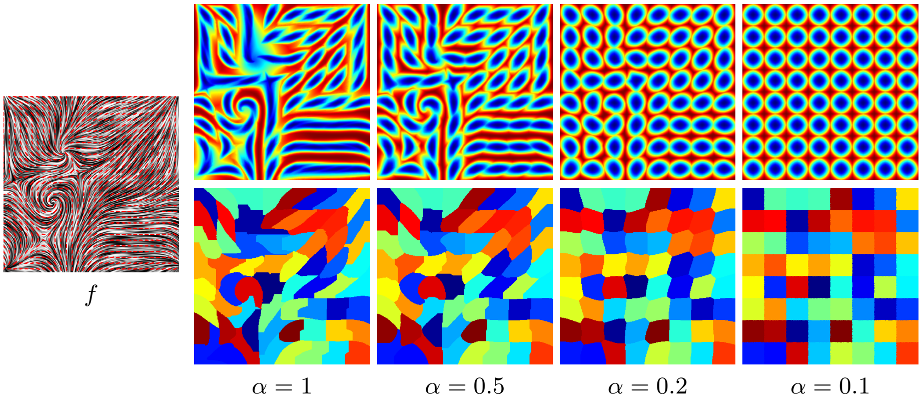


Fig. 4. Anisotropic distances (top row) and Voronoi diagrams (bottom row) with a decreasing anisotropy α. The metric tensor is computed using the structure tensor (8).

i.e. incident to only one edge (their corresponding Voronoi cells are adjacent to only one other cell). Also, some triangles can overlap. If the sampling $\mathcal{S}$ is in general position and dense enough, then one can prove that $\mathcal{D}_\mathcal{S}$ is a valid planar triangulation, see [1] for more details.

While the set of edges of $\mathcal{D}_\mathcal{S}$ forms a planar graph (straight line edges), its geometric realization connects points of $\mathcal{S}$ with geodesic paths. The geometric realization $\gamma_{i,j} \in \mathcal{P}(x_i, x_j)$, of a Delaunay edge $(x_i, x_j) \in \mathcal{D}_\mathcal{S}$, is defined as the union of the two geodesics joining the double point $w_{i,j}$ to x_i and x_j.

Numerical Computation. The Voronoi diagram (7) can be computed in parallel to the computation of the geodesic distance map $U_\mathcal{S}$. This requires to track the index of the closest point in $\mathcal{S}$ of any point in the front during the Fast Marching propagation. This can be done with any algorithm mentioned in Section 1. During this propagation, one can also keep track of double points and Voronoi vertices. For each Delaunay edge $(x_i, x_j) \in \mathcal{D}_\mathcal{S}$, its double point $w_{i,j}$ corresponds to the first meeting location of the two fronts emanating from both x_i and x_j. After the propagation, the geometric realization of each Delaunay edge (x_i, x_j) is extracted by solving two gradient descents, according to (5), to compute the two geodesics joining $w_{i,j}$ to x_i and to x_j.

All the geometric tools defined above can thus be extracted using a single front propagation, which requires $O(n \log(n))$ operations. Furthermore, when they have been computed for a set of points $\mathcal{S}$, they can be extended to $\mathcal{S} \cup \{x_{m+1}\}$ by a local propagation restricted to the cell $\mathcal{C}_{m+1}$, thus requiring on average $O(n \log(n)/m)$ operations.

Figure 4 shows examples of Voronoi diagrams for Riemannian metrics with a decreasing anisotropy. One can see how the Voronoi cells $\mathcal{C}_i$ are stretched along the main tensor direction e_1 for highly anisotropic metrics.

3 Application to Perceptual Grouping

Perceptual grouping is a curve reconstruction problem where one wants to extract a curve from an image containing a sparse set of curves, eventually embedded in noise. This problem is relevant both to model good continuation perception laws and to develop efficient edge detection methods (see for instance [11]). In this paper we restrict ourselves to the detection of a set of non-intersecting open or closed curves, although other kinds of topological or regularity constraints could be enforced.

Our algorithm extends the isotropic geodesic grouping method of Cohen [12] by designing a Riemannian metric that propagates the anisotropy of the sparse curves to the whole domain. This metric helps to disambiguate difficult situations where some curves are close from one to each other. This allows a better reconstruction with less user intervention. The idea of using anisotropic information to perform perceptual grouping is introduced in [13] where the completed contours are local minimizer of a saliency field. Many variational definitions of perceptual contours have been proposed using local regularity assumption, for instance with the elastica model of Mumford [14]. In contrast, our completed contours are anisotropic shortest paths that connect Riemannian Voronoi cells, thus being the global minimum of a length criterion.

Design of an anisotropic tensor field. The Riemannian metric H needs to be computed from the noisy input image $f:[0,1]^2 \rightarrow \mathbb{R}$. In order to compute robustly the local direction of the features, we use a local pooling of the gradient information that constitutes a sparse set of tensors with a confidence measure. This sparse tensor field is then integrated by diffusion into a dense field.

The local orientation of a feature around a pixel x is given by the vector orthogonal to the gradient $v(x) = (\nabla_x f)^{\perp}$, which is computed numerically with finite differences (using maybe some little smoothing to cancel noise). This local direction information can be stored in a rank-1 tensor $T_0(x) = v(x)v(x)^{\mathrm{T}}$. In order to evaluate the local anisotropy of the image, one needs to average this tensor

$$T(x) = T_0 * G_\sigma(x), \tag{8}$$

where the four entries of the tensor are smoothed against a Gaussian kernel G_σ of width $\sigma > 0$. The metric H corresponds to the so-called structure tensor, see for instance [15]. This local tensor T is able to extract both the local direction of edges and the local direction of textural patterns (see Figure 4, left).

The structure tensor field T defined by (8) gives a robust estimation of the local anisotropy only close to image features where the gradient is large. In homogeneous areas (typically outside the salient features), the tensor is nearly isotropic with small eigenvalues. In order to have a dense anisotropic field, one needs to extend the anisotropy over the whole domain using some kind of interpolation. This notion of interpolation of local orientations is similar to the computation of good continuation field, as studied for instance in stochastic completion fields [16] or tensor voting [17].

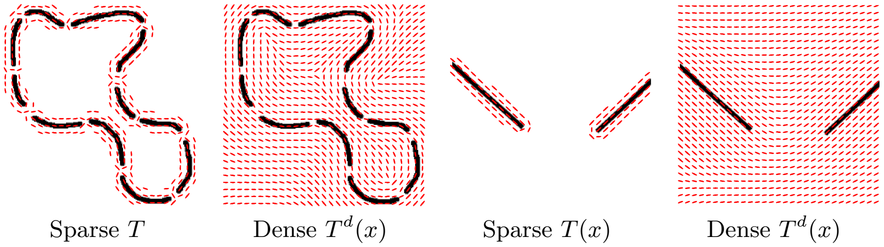


Fig. 5. Computation of a dense tensor field $T^d(x)$ from a sparse structure tensor $T(x)$.

In this paper, we propose a simple interpolation method that computes a dense tensor field with a linear diffusion outside a region of high confidence. The region of high saliency is computed by thresholding the anisotropy map α of T defined in (2)

$$\Omega_\alpha \overset{\text{def.}}{=} \{x \in \Omega \setminus \alpha(x) > 1 - \varepsilon\},$$

where ε is a small constant. In our numerical examples we choose $\varepsilon = 0.05$. The orientation of the tensors in Ω_α are computed with a high confidence, and in order to compute a dense tensor field, we compute the following steady state of a heat diffusion

$$\forall\, x \notin \Omega_\alpha, \quad \Delta T^d_{i,j}(x) = 0 \quad \text{and} \quad \forall\, x \in \Omega_\alpha, \quad T^d_{i,j}(x) = T_{i,j}(x),$$

where Δ is the Laplacian with Neumann reflecting conditions on the boundary $\partial\Omega$ and where $(T_{i,j})_{i,j=1,2}$ are the components of the structure tensor.

In order to turn the dense tensor field T^d into a Riemannian metric, we apply a non-linear mapping to its eigenvalues,

$$T^d(x) = \mu_1 e_1 {e_1}^{\mathrm{T}} + \mu_2 e_2 {e_2}^{\mathrm{T}} \quad \Longrightarrow \quad H(x) = \psi_1(\mu_1) e_1 {e_2}^{\mathrm{T}} + \psi_2(\mu_2) e_2 {e_2}^{\mathrm{T}}. \quad (9)$$

where ψ_i is a decreasing function. In this paper, we use $\psi_i(x) = (\eta + |x|)^{-1}$ for a small value of η. Figure 5 shows examples of the proposed method.

Anisotropic Perceptual Grouping. Our anisotropic grouping algorithm proceeds by computing a perceptual graph $\tilde{\mathcal{D}}_{\mathcal{S}}$ of a set of points $\mathcal{S}$ provided by the user. This perceptual graph is a sub-graph of the set of edges of $\mathcal{D}_{\mathcal{S}}$.

The use of the anisotropic metric H defined in (9) helps to reduce the user intervention by grouping curves that obey a good continuation property with respect to the anisotropic tensor field. This is performed by ordering the edges $(x_i, x_j) \in \mathcal{D}_{\mathcal{S}}$ with respect to their respective geodesic distance $d(x_i, x_j)$. The edges are progressively inserted as long as the reconstructed curves are closed or open but non-intersecting. This topological constraint is enforced by monitoring the current degree δ_i of each point, which should be smaller or equal to 2. The algorithm is detailed in Table 1.

```
1. Initialization: set $\tilde{\mathcal{D}}_\mathcal{S} \leftarrow \emptyset$, and $\Pi \leftarrow \mathcal{D}_\mathcal{S}$.
2. Select edge: set $(x_i, x_j) \leftarrow \operatorname{argmin}_{(x,y)\in\Pi} d(x,y)$. Remove it: $\Pi \leftarrow \Pi - \{(x_i, x_j)\}$.
3. Check topology: if $\delta_i < 2$ and $\delta_j < 2$, then update $\tilde{\mathcal{D}}_\mathcal{S} \leftarrow \tilde{\mathcal{D}}_\mathcal{S} \cup \{(x_i, x_j)\}$ and set
   $\delta_j \leftarrow \delta_j + 1$ and $\delta_i \leftarrow \delta_j + 1$.
4. Stop: while $\Pi \neq \emptyset$, go back to 2.
```

Table 1: Anisotropic perceptual grouping algorithm.

Figure 6 compares the results of perceptual grouping using an isotropic metric (which is equivalent to the algorithm developed in [12]) to our algorithm that uses the dense tensor field T^d. Contrary to the isotropic method, our anisotropic metric enables a correct grouping that obeys a good continuation property.

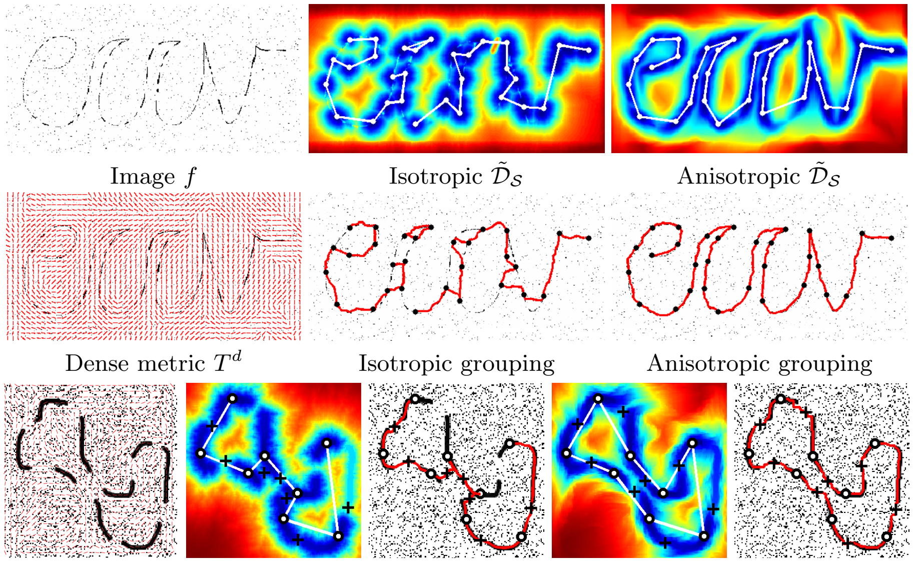


Image f Isotropic $\tilde{\mathcal{D}}_\mathcal{S}$ Anisotropic $\tilde{\mathcal{D}}_\mathcal{S}$

Dense metric T^d Isotropic grouping Anisotropic grouping

f and T^d Isotropic $\tilde{\mathcal{D}}_\mathcal{S}$ Grouping Anisotropic $\tilde{\mathcal{D}}_\mathcal{S}$ Grouping

Fig. 6. Comparison of isotropic and anisotropic Riemannian metrics for perceptual grouping.

4 Application to Meshing of Planar Domains

Planar domain meshing requires to build a good quality triangulation of a given domain. Triangles with anisotropic shape and varying sizes are highly desirable because of their capability to represent efficiently functions with directional singularities that one encounters for instance in parabolic PDE's near shocks.

This section proposes a new anisotropic meshing algorithm that conforms the shape and size of the triangles to the tensor field of a Riemannian metric.

Classical planar domain meshing algorithms are based on Euclidean Delaunay triangulation. They proceed by progressively inserting Voronoi vertices, see for instance [18, 19]. These points are inserted in order to split triangles that are poorly shaped, and also to ensure a minimum size of the triangles.

Anisotropic meshes can be built using a local modification of the metric [20] or anisotropic elastic forces [21], bubble packing [22], simplified anisotropic metric [3]. This latter approach is used in conjunction with Ruppert's Delaunay refinement algorithm to provide anisotropic triangulation with guaranties on the aspect ratio of the triangles. This algorithm is extended to domains with curves by [23], and an alternative construction of the anisotropic Voronoi diagram is proposed in [24]. The simplified distance has been applied to image sampling [25], optimal samples placement with centroidal tessellations [26] and surface remeshing [27].

n this paper, we propose a new Delaunay refinement algorithm that extends the algorithm of Ruppert [18], and the one of Labelle and Shewchuck [3], with anisotropic geodesic distances. It also extends the isotropic farthest point seeding strategy of [28] with anisotropic metrics and domains with arbitrary boundaries.

As these algorithms, our anisotropic meshing algorithm proceeds by iteratively inserting points to an already computed set of points $\mathcal{S}$. The inserted points are the Voronoi vertices of the anisotropic geodesic Voronoi diagram $\mathcal{V}_\mathcal{S}$. The resulting mesh is the Delaunay complex $\mathcal{D}_\mathcal{S}$ dual to $\mathcal{V}_\mathcal{S}$. The main difficulty is that $\mathcal{D}_\mathcal{S}$ is not necessarily a valid planar triangulation if $\mathcal{S}$ is not dense enough. As described in Section 2, this is due to the presence of some isolated points or some overlapping triangles. As in [3] and [24], the algorithm automatically add points to overcome these problems and to guaranty a valid planar triangulation.

At least one point of $\mathcal{S}$ is located on each boundary curve, and these boundary points segment $\partial\Omega$ as a set of sub-curves

$$\partial\Omega = \bigcup_{i,j} \theta_{i,j} \quad \text{with} \quad \begin{cases} \theta_{i,j} \in \mathcal{P}(x_i, x_j), \\ \theta_{i,j} \cap \mathcal{S} = \{x_i, x_j\}, \end{cases}$$

(one can have $x_i = x_j$ if there is only one point on a curve).

In order to compute an anisotropic triangular mesh with triangles of high quality with respect to the local metric, one needs to iteratively insert the Voronoi vertex $\mathcal{V}_{i,j,k}$ dual to the Delaunay triangle (x_i, x_j, x_k) with the smallest circumradius to shortest edge ratio

$$\rho(\mathcal{V}_{i,j,k}) = \frac{U_\mathcal{S}(\mathcal{V}_{i,j,k})}{\min(d(x_i,x_j), d(x_j,x_k), d(x_k,x_i))},$$

which is a quantity computed for each Voronoi vertex in parallel to the FM propagation. In the Euclidean domain, a triangle (x_i, x_j, x_k) with a high value of $\rho(\mathcal{V}_{i,j,k})$ is badly shaped since its smallest angle is close to 0. As explained in [3], this property extends to an anisotropic metric H if angles are measured using the inner product defined by H.

One can also consider the farthest point strategy to refine the mesh, e.g. the Voronoi vertex $\mathcal{V}_{i,j,k}$ with the greatest distance $U_{\mathcal{S}}(\mathcal{V}_{i,j,k})$ is iteratively inserted.

The major obstacle to these insertions (points to get a planar triangulation and Voronoi vertices) is that the boundary $\partial\Omega$ should always be represented in the triangulation. This can be broken if a point x_k is located too close to the boundary. A boundary sub-curve $\theta_{i,j} \in \partial\Omega$ is said to be encroached by a point $x_k \in \mathcal{S}$, $x_k \notin \partial\Omega$, if the Voronoi cell $\mathcal{C}_k$ of x_k satisfies $\mathcal{C}_k \cap \partial\Omega \neq \emptyset$. Thus, the edge (x_i, x_j) cannot be part of the Delaunay triangulation, and it is automatically split by the algorithm by inserting the point $y_{i,j} \in \theta_{i,j} \cap \mathcal{C}_k$ as close to the midpoint of $\theta_{i,j}$ as possible. Similarly, Voronoi vertices are not added if they encroach any boundary sub-curve (the sub-curve is subdivided instead).

Table 2 details this algorithm. A bound $\rho^\star$ on ρ enforces the refinement to reach some quality criterion, while a bound $U^\star$ enforces a uniform refinement to match some desired triangle density.

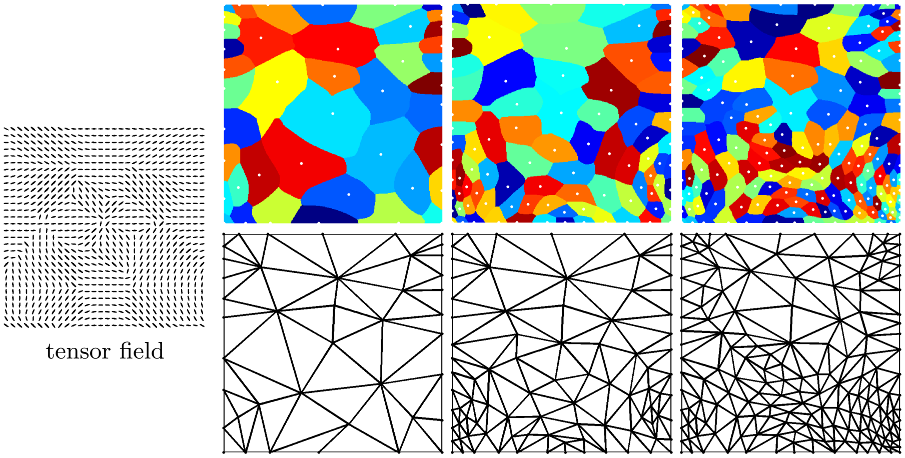
1. *Initialization:* set $\mathcal{S}$ with at least one point on each curve of Ω, compute $U_{\mathcal{S}}$ with a Fast Marching.
2. *Boundary enforcement:* while it exists $\theta_{i,j} \subset \partial\Omega$ encroached by some $x_k \in \mathcal{S}$, subdivide: $\mathcal{S} \leftarrow \mathcal{S} \cup \{y_{i,j}\}$ (see text). Update $U_{\mathcal{S}}$ with a local Fast Marching.
3. *Triangulation enforcement:* while it exists $(x_i, x_j) \in \mathcal{D}_{\mathcal{S}}$ with x_i or x_j isolated, insert $\underset{x \in \mathcal{C}_i \cap \mathcal{C}_j}{\operatorname{argmax}}\, U_{\mathcal{S}}(x)$. Update $U_{\mathcal{S}}$ with a local Fast Marching.
4. *Select point:* $s^\star \leftarrow \underset{s \in \Sigma_{\mathcal{S}} - \partial\Omega}{\operatorname{argmin}}\, \rho(s)$ (or $s^\star \leftarrow \underset{s \in \Sigma_{\mathcal{S}} - \partial\Omega}{\operatorname{argmax}}\, U_{\mathcal{S}}(s)$).
   – If in $\mathcal{S} \cup \{s^\star\}$, $s^\star$ encroaches some $\theta_{i,j} \subset \partial\Omega$, subdivide: $\mathcal{S} \leftarrow \mathcal{S} \cup \{y_{i,j}\}$.
   – Otherwise, add it: $\mathcal{S} \leftarrow \mathcal{S} \cup s^\star$.
   Update $U_{\mathcal{S}}$ with a local Fast Marching.
5. *Stop:* while $\rho(s^\star) > \rho^\star$ or $U_{\mathcal{S}}(s^\star) > U^\star$, go back to 2.
6. *Triangulation enforcement:* if it exists two adjacent triangles that overlap, add the dual Voronoi vertex of one of them to $\mathcal{S}$, update $U_{\mathcal{S}}$ with a local Fast Marching, and go back to 2.

Table 2: Anisotropic planar domain meshing algorithm.

Figure 7 shows examples of meshing with a decreasing value of the parameter $\rho^\star$. The number of points is more and more important while $\rho^\star$ is decreasing. Figure 8 shows results of meshing with the farthest point strategy (parameter U^*) for several tensor fields.

The tensor field can be given by the user (top row) or computed from some background image (using the structure tensor, equation (8)), which can be interesting to perform image approximation by linear splines on triangles.

Figure 9 shows that our algorithm can handle domain Ω or arbitrary complexity. Here the tensor field is computed to follow the medial axis of the domain, thus stretching the triangles in an appropriate direction.



tensor field

41 points inserted 75 points inserted 150 points inserted

Fig. 7. Meshing of a square with an increasing number of points, which is controlled by the parameter $\rho^\star$. The initial point set $\mathcal{S}$ is the four corner points.

Conclusion

This paper has detailed how several notions from computational geometry extend seamlessly to the geodesic setting. This allows to incorporate important knowledge about the directionality of the features to solve more efficiently problems in computer vision and graphics. We explore two particular applications, perceptual grouping and domain meshing, where this anisotropy allows to overcome several limitations of previous approaches.

References

1. Leibon, G., Letscher, D.: Delaunay triangulations and Voronoi diagrams for Riemannian manifolds. In: Symp. on Computational Geometry, ACM (2000) 341–349
2. Frey, P.J., George, P.L.: Mesh generation. Application to finite elements. 2nd edn. Wiley (2008)
3. Labelle, F., Shewchuk, J.R.: Anisotropic voronoi diagrams and guaranteed-quality anisotropic mesh generation. In: Symp. on Computational Geometry. (2003) 191–200
4. Crandall, M.G., Ishii, H., Lions, P.L.: User's guide to viscosity solutions of second order partial differential equations. Bull. Amer. Math. Soc. **27** (1992) 1–67
5. Cohen, L.D., Kimmel, R.: Global Minimum for Active Contour models: A Minimal Path Approach. International Journal of Computer Vision **24**(1) (Aug. 1997) 57–78
6. Sethian, J.: Level Sets Methods and Fast Marching Methods. 2nd edn. Cambridge University Press (1999)
7. Kimmel, R., Sethian, J.: Computing Geodesic Paths on Manifolds. Proc. Natl. Acad. Sci. **95**(15) (1998) 8431–8435

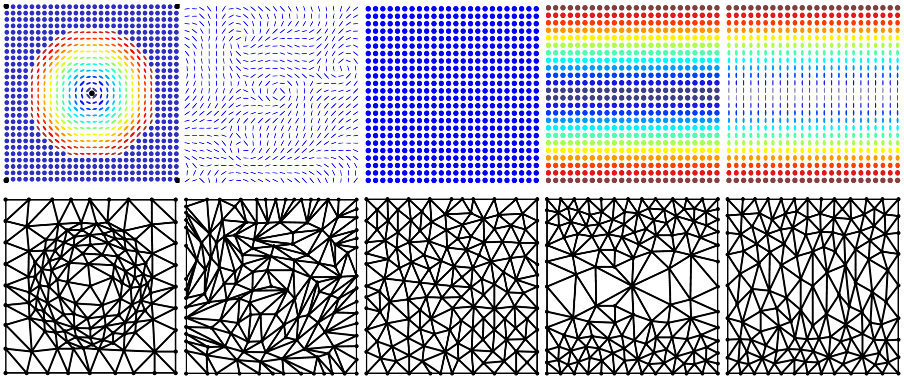


Fig. 8. Meshing of a square with the farthest point strategy, which is controlled by the parameter U^*.

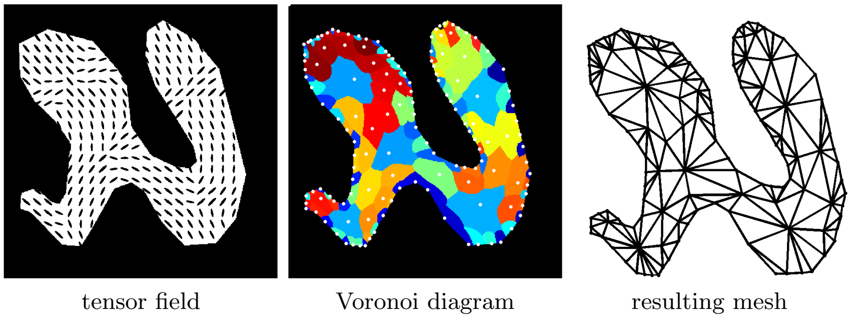


Fig. 9. Meshing of a complex domain.

8. Spira, A., Kimmel, R.: An efficient solution to the eikonal equation on parametric manifolds. Interfaces and Free Boundaries **6**(3) (2004) 315–327
9. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Weighted distance maps computation on parametric three-dimensional manifolds. Journal of Computational Physics, accepted (2007)
10. Prados, E., Lenglet, C., Pons, J.P., Wotawa, N., Deriche, R., Faugeras, O.D., Soatto, S.: Control theory and fast marching methods for brain connectivity mapping. In: Proc. CVPR 2006. (2006)
11. Ren, X., Fowlkes, C.C., Malik, J.: Learning probabilistic models for contour completion in natural images. Int. Journal of Computer Vision **77** (2008) 47–63
12. Cohen, L.D.: Multiple contour finding and perceptual grouping using minimal paths. Journal of Mathematical Imaging and Vision **14**(3) (2001) 225–236
13. Guy, G., Medioni, G.: Inferring global perceptual contours from local features. International Journal of Computer Vision **20**(1-2) (1996) 113–133
14. Mumford, D.: Elastica and computer vision. In: Algebraic geometry and its applications. 491–506
15. Kothe, U.: Edge and junction detection with an improved structure tensor. In: Proc. DAGM03. (2003) 25–32

16. Williams, L., Jacobs, D.W.: Stochastic completion fields: a neural model of illusory contour shape and salience. Neural Comput. **9**(4) (1997) 837–858
17. Medioni, G., Lee, M.S., Tang, C.K.: A Computational Framework for Segmentation and Grouping. Elsevier (2000)
18. Ruppert, J.: A Delaunay refinement algorithm for quality 2-dimensional mesh generation. J. Algorithms **18**(3) (1995) 548–585
19. Chew, L.P.: Guaranteed-quality mesh generation for curved surfaces. In: Proc. of SCG '93, New York, NY, USA, ACM (1993) 274–280
20. Borouchaki, H., George, P., Mohammadi, B.: Delaunay mesh generation governed by metric specifications. Part I. algorithms. Finite Elem. Anal. Des. **25**(1-2) (1997) 61–83
21. Bossen, F.J., Heckbert, P.S.: A pliant method for anisotropic mesh generation. In: 5th Intl. Meshing Roundtable. (October 1996) 63–74
22. Yamakawa, S., Shimada, K.: High quality anisotropic tetrahedral mesh generation via ellipsoidal bubble packing. In: IMR. (2000) 263–274
23. Yokosuka, Y., Imai, K.: Guaranteed-quality anisotropic mesh generation for domains with curves. In: Proc. of EWCG'06. (2006)
24. Boissonnat, J.D., Wormser, C., Yvinec, M.: Anisotropic diagrams: Labelle shewchuk approach revisited. In: CCCG. (2005) 266–269
25. Feng, Z., Hotz, I., Hamann, B., Joy, K.I.: Anisotropic noise samples. IEEE Transactions on Visualization and Computer Graphics **14**(2) (2008) 342–354
26. Du, Q., Wang, D.: Anisotropic centroidal Voronoi tessellations and their applications. SIAM Journal on Scientific Computing **26**(3) (May 2005) 737–761
27. Valette, S., Chassery, J., Prost, R.: Generic remeshing of 3d triangular meshes with metric-dependent discrete Voronoi diagrams. IEEE Trans Visu Comp Graph **14**(2) (2008) 369–381
28. Peyré, G., Cohen, L.D.: Geodesic remeshing using front propagation. Int. J. Comput. Vision **69**(1) (2006) 145–156