Anisotropic Geodesics for Perceptual Grouping and Domain Meshing

Sébastien Bougleux, Gabriel Peyré, Laurent D. Cohen

Anisotropic Geodesics for Perceptual Grouping and Domain Meshing

Sébastien Bougleux, Gabriel Peyré and Laurent Cohen

Ceremade, Université Paris-Dauphine, 75775 Paris Cedex 16, France
{bougleux,peyre,cohen}@ceremade.dauphine.fr,
WWW home page:
http://www.ceremade.dauphine.fr/{∼bougleux/,∼peyre/,∼cohen/}

Abstract. This paper shows how computational Riemannian manifold can be used to solve several problems in computer vision and graphics. Indeed, Voronoi segmentations and Delaunay graphs computed with geodesic distances are shaped according to the anisotropy of the metric. A careful design of a Riemannian manifold can thus help to solve some important difficulties in computer vision and graphics. The first contribution of this paper is thus a detailed exposition of Riemannian metrics as a tool for computer vision and graphics. The second contribution of this paper is the use of this new framework to solve two important problems 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 curves. Our anisotropic grouping algorithm works over a Riemannian metric that propagates the direction of a sparse set of noisy incomplete curves over the whole domain. The proposed method prunes the Delaunay graph in order to correctly link together salient features in the image. The second problem studied is planar domain meshing, where one should build a good quality triangulation of a given domain. Our anisotropic meshing algorithm is a geodesic Delaunay refinement method that exploits a Riemannian metric in order to locally impose the orientation and aspect ratio of the created triangles.

1 Geodesic Distance over Riemannian Manifold

Various important problems in computer graphics and computer vision require the integration of a local anisotropy over a complex 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. In this paper, we focus on two representative problems in these fields: perceptual grouping of salient features and meshing of a complicated planar domain.

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.

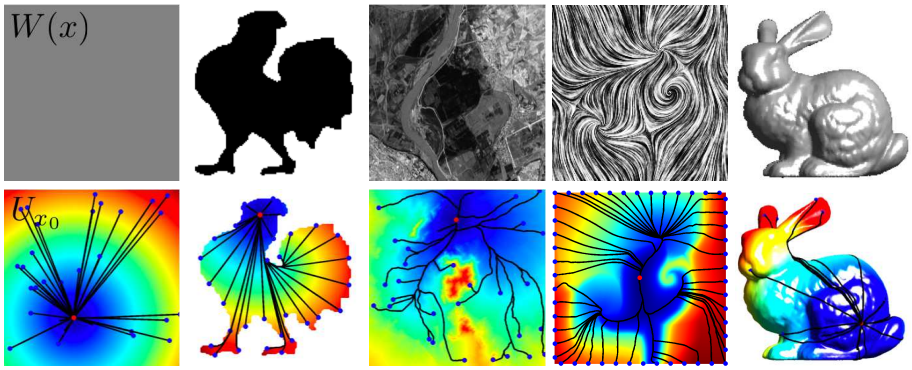


Fig. 1. *Examples of Riemannian metrics (top row) and geodesic distances and curves (bottom row). The blue/red colormap indicates the geodesic distance to the starting point. From left to right: euclidean ($H(x) = \mathrm{Id}_2$ restricted to $\Omega = [0,1]^2$), planar domain ($H(x) = \mathrm{Id}_2$ restricted to $\Omega \neq [0,1]^2$), isotropic ($H(x) = W(x)\mathrm{Id}_2$ with W computed from the image), Riemannian manifold metric ($H(x)$ is the structure tensor of the image, see equation (8)) and 3D surface ($H(x)$ corresponds to the first fundamental form).*

Figure 1 shows some examples of Riemannian metrics frequently encountered in computer vision and graphics. This section reviews basis facts about Riemannian manifolds and explains how geodesic distances can be computed efficiently with Fast Marching methods.

1.1 Riemannian Metric and Geodesic Distance

This paper considers 2D Riemannian manifolds that are defined over a compact planar domain $\Omega \subset \mathbb{R}^2$. 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

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

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 f to process. In computer graphics and numerical analysis, one often deals with complicated planar domains Ω with holes and corners. Figure 1 shows some examples of Riemannian metrics.

An important issue is thus to design, for a specific application, a tensor field $H(x)$ that encodes the important information about the problem to solve. Sections 3 and 4 show computer vision and graphics applications where the tensor field is derived either from a background image or from some user input.

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

$$H(x) = \lambda_1(x)e_1(x)e_1(x)^{\mathrm{T}} + \lambda_2(x)e_2(x)e_2(x)^{\mathrm{T}} \quad \text{with} \quad 0 < \lambda_1 \leqslant \lambda_2, \qquad (1)$$

and e_1, e_2 are two 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 (to be defined next) tends to be tangent to the direction field e_1.

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

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

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

1.2 Geodesic Distances and Shortest Paths

The geodesic distance between two points is defined as

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

where $\mathcal{P}(x, y)$ denotes the set of piecewise smooth curves joining x and y

$$\mathcal{P}(x, y) \overset{\text{def.}}{=} \{\gamma \setminus \gamma(0) = x \quad \text{and} \quad \gamma(1) = y\}.$$

The distance d turns the domain Ω into a metric space. By carefully designing a local Riemannian metric, one can create global interactions within the domain that are crucial to solve computer vision and graphics problems such as perceptual grouping or planar domain meshing.

A shortest path γ between two points $(x, y) \in \Omega^2$ according to the Riemannian metric is called a geodesic. It satisfies $L(\gamma) = d(x, y)$. If the metric H is well chosen, then geodesic curves can be used to follow salient features in images.

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 define the distance map to this set

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

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 PDE

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

where $\|v\|_A = \sqrt{v^{\mathrm{T}} A v}$. This classical result allows to replace the optimization problem (3) that defines geodesic distances by a partial differential equation.

Once the distance map $U_{\mathcal{S}}$ has been computed by solving the Eikonal equation (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

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

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

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

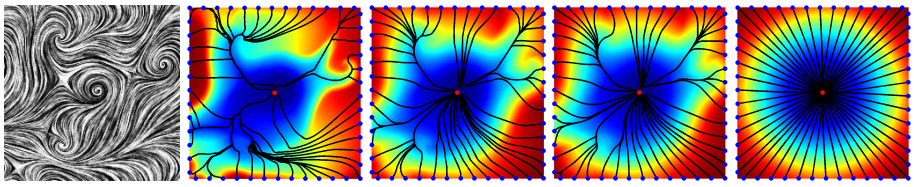


Fig. 2. *Examples of geodesics for a tensor metric with an increasing anisotropy α (see equation (2) for a definition of this parameter). The tensor field $H(x)$ is computed from the structure tensor of f as defined in equation (8), its eigenvalues fields $\lambda_i(x)$ are then modified to impose the anisotropy α.*

For the particular case of an isotropic metric $H(x) = W(x)^2 \mathrm{Id}_x$, 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 [1].

1.3 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 algorithm, introduced by Sethian [2] is a numerical procedure to efficiently solve in $O(n \log(n))$ operations the discretization of equation (6) in the isotropic case. Several extensions of the Fast Marching have been proposed in order to solve equation (4) for a generic metric, see for instance Kimmel and Sethian [3] for triangulated meshes and Spira and Kimel [4], Bronstein et al. [5] for parametric manifolds.

We use the Fast Marching method developed by Prados et al. [6], which is a numerical scheme to compute the geodesic distance over a generic parameteric Riemannian manifold in 2D and 3D in $O(n \log(n))$ operations. As any Fast Marching method, it computes the distance $U_{\mathcal{S}}$ by progressively propagating a front, starting from the initial points in $\mathcal{S}$. Figure 3 shows an example of Fast Marching computation with an anisotropic metric. The front propagates faster

along the direction of the texture. This is because the Riemannian tensor is computed following equation (8) in order for the principal direction e_1 to align with the texture patterns.

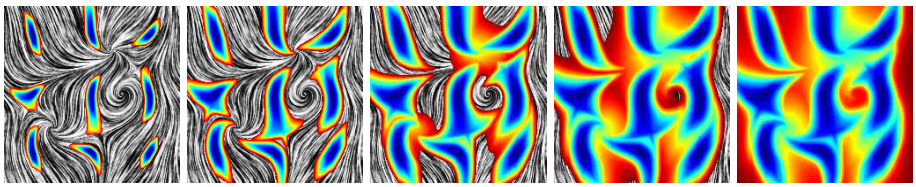


Fig. 3. *Examples of anisotropic front propagation (from 9 starting points). The colormap indicates the values of the distance functions at a given iteration of the algorithm. The metric is computed using the structure tensor, equation* (8)*, of the texture f shown in the background.*

2 Computational Geometry with Riemannian Metric

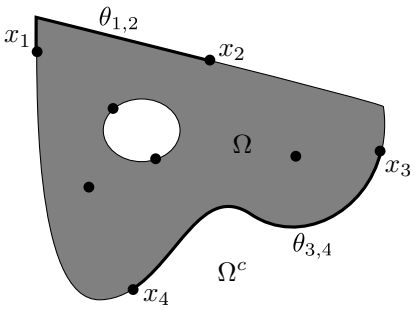
This section, shows how several tools from computational geometry extend to the setting of a Riemannian metric.

Starting from a set of points $\mathcal{S} = \{x_i\}_{i=1}^m$, one can define graphs and triangulations that reflect the geometry of the Riemannian manifold. These points and the corresponding graphs are the basic building blocks of the algorithms for perceptual grouping and planar domain meshing.

In the following, the boundary $\partial\Omega$ of Ω is assumed to be a set of closed smooth curves. At least one point of $\mathcal{S}$ is located on each 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).



2.1 Delaunay and Voronoi Graphs

The segmentation of the domain Ω in Riemannian Voronoi cells is

$$\Omega = C_0 \bigcup_{x_i \in \mathcal{S}} \mathcal{C}_i \quad \text{where} \quad \mathcal{C}_i = \{x \in \Omega \setminus \forall\, j \neq i, \quad d(x_i, x) \leqslant d(x_j, x)\}\,. \tag{7}$$

The outer Voronoi cell is defined as $\mathcal{C}_0 = \text{Closure}(\Omega^c)$.

The Delaunay graph $\mathcal{D}_\mathcal{S}$ of $\mathcal{S}$ is a graph where two points are connected if their respective Voronoi cells are adjacent

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

To each Delaunay edge $(x_i, x_j) \in \mathcal{D}_\mathcal{S}$ corresponds a double point $w_{i,j}$, which is the closest point to x_i and x_j on the common Voronoi cell boundary

$$w_{i,j} = \operatorname*{argmin}_{x \in \mathcal{C}_i \cap \mathcal{C}_j} d(x, x_i).$$

The Delaunay graph $\mathcal{D}_\mathcal{S}$ is a planar graph, where each Delaunay edge $(x_i, x_j) \in \mathcal{D}_\mathcal{S}$ has a geometric realization that is defined as

- if one of x_i or x_j is not on $\partial\Omega$, the curve (x_i, x_j) is the union of the two geodesics joining the double point $w_{i,j}$ to x_i and x_j,
- if $x_i \in \partial\Omega$ and $x_j \in \partial\Omega$, the curve (x_i, x_j) is the portion of the boundary $\theta_{i,j} \subset \partial\Omega$.

If the sampling $\mathcal{S}$ of Ω is dense enough, then one can prove that $\mathcal{D}_\mathcal{S}$ is a valid planar triangulation, see [7].

A triple point $s_{i,j,k}$ is located at the intersection of three adjacent Voronoi cells

$$s_{i,j,k} \in \mathcal{C}_i \cap \mathcal{C}_j \cap \mathcal{C}_k.$$

The set of triple points of $\mathcal{S}$ is denoted as $\Sigma_\mathcal{S}$. For a generic set of points $\mathcal{S}$, these triple points are isolated. Saddle points $s_{i,j,k}$ where one of the index i, j or k is 0 are located on the boundary $\partial\Omega$ of the domain.

2.2 Numerical Computation

The Voronoi segmentation (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 1.3.

During this propagation, one can keep track of the double points $w_{i,j}$ for each Delaunay edge $(x_i, x_j) \in \mathcal{D}_\mathcal{S}$. Such a point corresponds to the first meeting location of the two fronts emanating from both x_i and x_j. After the propagation is finished, the Delaunay edge curve (x_i, x_j) is extracted by solving two gradient descents, equation (5), to compute the two geodesics joining $w_{i,j}$ to x_i and x_j.

All the geometric tools (Delaunay graph, Voronoi cells, double and triple points) 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 segmentation 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.

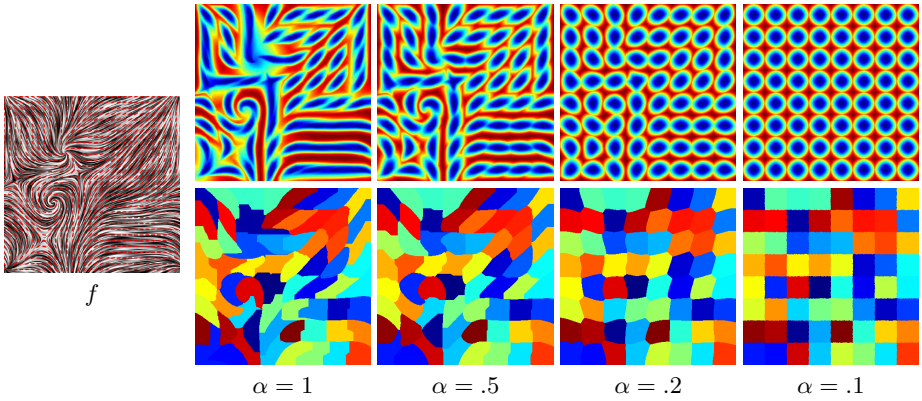


Fig. 4. *Examples of anisotropic distances (top row) and Voronoi diagrams (bottom row) with an decreasing anisotropy α. The metric tensor is computed using the structure tensor, equation* (8).

3 Application to Perceptual Groupping

Perceptual grouping is a curve reconstruction problem where one wants to extract a curve from an image containing a sparse set of curves embedded in noise. This problem is relevant both to model good continuation perception laws and to develop efficient edge detection methods. 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 [8] by designing a Riemannian metric that propagates the anisotropy of the sparse curves to the whole domain. This metric helps to disambiguates 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 [9] where the completed contours are local minimizers of a saliency field. Many variational definitions of perceptual contours have been proposed using local regularity assumption, for instance with the ellastica model of Mumford [10]. In contrast, our completed contours are anisotropic shortest paths that connect Riemannian Voronoi cells, thus being the global minimum of a length criterion.

3.1 Design of an Anisotropic Tensor Field

The Riemannian metric $H(x)$ needs to be computed from the noisy input image f. 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.

Structure Tensor. 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 4 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 [11]. 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). Another option, that we do not pursue here, is to use the square of the Hessian matrix of f instead of the structure tensor.

Estimation of a dense tensor field. The structure tensor field $T(x)$ defined in (8) gives a robust estimation of the local anisotropy only close to image features, where the gradient is large. In homogenous 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 orientation is similar to the computation of good continuation field, as studied for instance in stochastic completion fields [12] or tensor voting [13].

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 $T(x)$ defined in equation (2)

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

where ε is a small constant, in the numerical examples we choose $\varepsilon = .05$. 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,

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

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 computation of dense tensor fields.

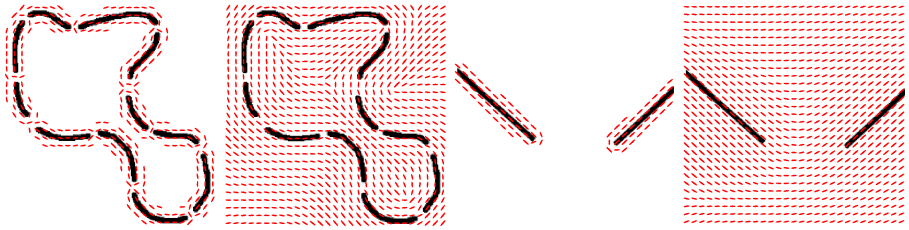


Fig. 5. *Computation of a dense tensor field $T^d(x)$ from a sparse structure tensor $T(x)$.*

3.2 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 Delaunay graph $\tilde{\mathcal{D}}_{\mathcal{S}} \subset \mathcal{D}_{\mathcal{S}}$.

The use of the anisotropic metric $H(x)$ defined in (9) helps to reduce the user intervention to a minimum by grouping together only 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 at the corresponding 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 $(x_i, x_j) \longleftarrow \underset{(x,y)\in\Pi}{\operatorname{argmin}}\, d(x, y)$. 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 [8]) to our algorithm that uses the dense tensor field T^d. The isotropic method fails because closed curves are connected regardless of their relative orientation. In contrast, our anisotropic metric enables a correct grouping of curves that obey a good continuation property.

4 Application to Meshing of Planar Domains

Planar domain meshing. Planar domain meshing requires to build a good quality triangulation of a given domain. Triangles with anisotropic shape and

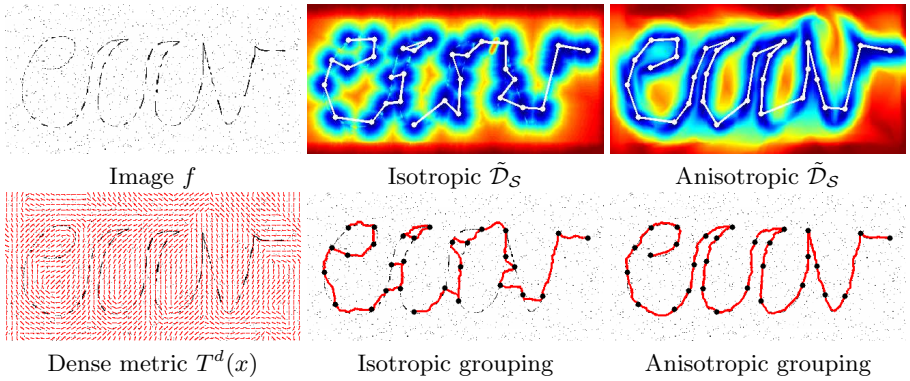


Fig. 6. *Comparison of isotropic and anisotropic Riemannian metric for perceptual grouping.*

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 triple points, which are circumcenters of Delaunay triangles, see for instance [14, 15]. 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 [16] or anisotropic elastic forces [17] and bubble packing [18]. In order to incorporate global constraints within a provably correct Delaunay refinement scheme, Labelle and Shewchuk propose in [19] to use the anisotropic distance $\tilde{d}(x,y)^2 = (x-y)^{\mathrm{T}} H(x)(x-y)$. This is a simplified anisotropic metric not based on shortest paths. Several issues arise with such a definition. It is not symmetric and the Voronoi diagram might not be connected. However it can be 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 in 3D by [20] and to domains with curves by [21]. The simplified distance $\tilde{d}$ has been applied to image sampling [22], optimal samples placement with centroidal tesselations [23] and surface remeshing [24].

Anisotropic geodesic meshing. We propose a new Delaunay refinement algorithm that extends the algorithm of Ruppert [14] with Riemannian metrics. It extends the isotropic farthest point seeding strategy of [25] with anisotropic metrics and domains with arbitrary boundaries.

Our anisotropic meshing algorithm proceeds by iteratively inserting a triple point $s_{i,j,k} \in \Sigma_{\mathcal{S}}$ to an already computed set of points $\mathcal{S}$. In order to compute an anisotropic mesh with triangles of high quality with respect to the local metric, one inserts $s_{i,j,k}$ for a Delaunay triangle (x_i, x_j, x_k) with the smallest

circumradius to shortest edge ratio

$$\rho(s_{i,j,k}) = \frac{d(s_{i,j,k}, x_i)}{\min(d(x_i, x_j), d(x_j, x_k), d(x_k, x_i))},$$

which is a quantity computed for each triple point in parallel to the Fast Marching propagation. In the Euclidean domain, a triangle (x_i, x_j, x_k) with a low value of $\rho(s_{i,j,k})$ is badly shaped since its smallest angle is close to 0 . As explained in [19], this property extends to an anisotropic metric $H(x)$ if angles are measured using the inner product defined by $H(x)$.

The major obstacle to the removing of badly shaped triangles is that the boundary $\partial\Omega$ should always be part of the Delaunay graph. This can be broken if a point x_k is located too close to the boundary. A boundary sub-curve $\theta_{i,j}$ is said to be encroached by $x_k \in \mathcal{S}$ if it exists a triple point $w_{i,k,0} \in \theta_{i,j}$, see figure 7. Such an encroached edge cannot be part of the Delaunay triangulation, and is automatically split by the algorithm by inserting a mid point. Similarly, triple points are not added if they encroach any boundary sub-curve (the sub-curve is subdivided instead).

Another difficulty is that the Delaunay graph $\mathcal{D}_\mathcal{S}$ of $\mathcal{S}$ is not necessarily a valid triangulation if the sampling $\mathcal{S}$ is not dense enough, see [7]. This is because of some isolated point, that is connected to only one other point of $\mathcal{S}$ in $\mathcal{D}_\mathcal{S}$. The algorithm automatically add points on the Voronoi cell boundary of such a point.

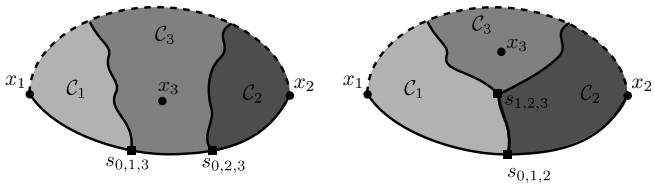


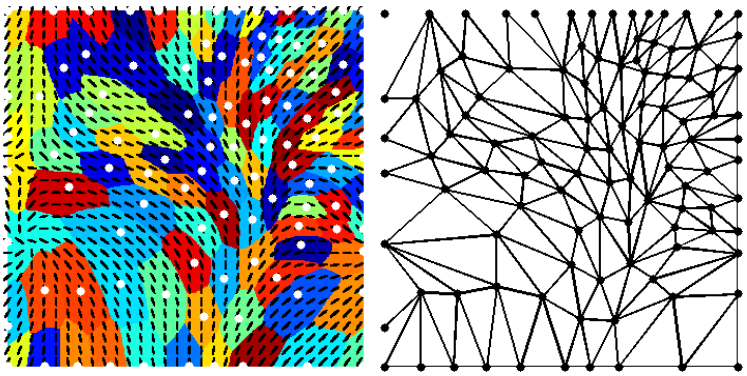
Fig. 7. *Left: the vertex x_3 encroaches the boundary curve $\theta_{1,2}$. Right: the vertex x_3 does not encroach anymore because (x_1, x_2) is a Delaunay edge.*

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: $\mathcal{S} \leftarrow \mathcal{S} \cup \operatorname{argmax}_{x \in \theta_{i,j}} U_{\mathcal{S}}(x)$. 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 $w^* = \operatorname{argmax}_{w \in C_i \cap C_j} d(x_i, w)$.
4. *Select point:* $s^\star \leftarrow \operatorname{argmin}_{s \in \Sigma_{\mathcal{S}} - \partial\Omega} \rho(s)$.
   – If in $\mathcal{S} \cup \{s^\star\}$, $s^\star$ encroaches some $\theta_{i,j} \subset \partial\Omega$, subdivide: $\mathcal{S} \leftarrow \mathcal{S} \cup \operatorname{argmax}_{x \in \theta_{i,j}} U_{\mathcal{S}}(x)$.
   – 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.

Table 2: Anisotropic planar domain meshing algorithm.

Figure 8 show an example of anisotropic meshing, where the user controls the shape of the triangle by designing the tensor field.



Voronoi and metric Delaunay triangulation

Fig. 8. *Example of anisotropic meshing with a synthetic tensor field.*

Conclusion

This paper has detailed how several notions from computational geometry extend seamlessly to the geodesic setting. This allows to incorporate some important knowledges 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. Cohen, L.D., Kimmel, R.: Global Minimum for Active Contour models: A Minimal Path Approach. International Journal of Computer Vision **24** (1997) 57–78
2. Sethian, J.: Level Sets Methods and Fast Marching Methods. 2nd edn. Cambridge University Press (1999)
3. Kimmel, R., Sethian, J.: Computing Geodesic Paths on Manifolds. Proc. Natl. Acad. Sci. **95** (1998) 8431–8435
4. Spira, A., Kimmel, R.: An efficient solution to the eikonal equation on parametric manifolds. Interfaces and Free Boundaries **6** (2004) 315–327
5. A. M. Bronstein, M. M. Bronstein, R.K.: Weighted distance maps computation on parametric three-dimensional manifolds. Journal of Computational Physics, accepted (2007)
6. 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)
7. Leibon, G., Letscher, D.: Delaunay triangulations and Voronoi diagrams for Riemannian manifolds. In: Proc. of SCG '00, New York, NY, USA, ACM (2000) 341–349
8. Cohen, L.: Multiple Contour Finding and Perceptual Grouping Using Minimal Paths. Journal of Mathematical Imaging and Vision **14** (2001) 225–236
9. Guy, G., Medioni, G.: Inferring global perceptual contours from local features. Int. J. Comput. Vision **20** (1996) 113–133
10. Mumford, D.: Elastica and computer vision. In C. L. Bajaj (Ed.), Algebraic geometry and its applications (1994) 491–506
11. Kothe, U.: Edge and junction detection with an improved structure tensor. In: Proc. DAGM03. (2003) 25–32
12. Williams, L., Jacobs, D.W.: Stochastic completion fields: a neural model of illusory contour shape and salience. Neural Comput. **9** (1997) 837–858
13. Medioni, G., Lee, M.S., Tang, C.K.: A Computational Framework for Segmentation and Grouping. Elsevier (2000)
14. Ruppert, J.: A Delaunay refinement algorithm for quality 2-dimensional mesh generation. J. Algorithms **18** (1995) 548–585
15. Chew, L.P.: Guaranteed-quality mesh generation for curved surfaces. In: Proc. of SCG '93, New York, NY, USA, ACM (1993) 274–280
16. Borouchaki, H., George, P., Mohammadi, B.: Delaunay mesh generation governed by metric specifications. Part I. algorithms. Finite Elem. Anal. Des. **25** (1997) 61–83
17. Bossen, F.J., Heckbert, P.S.: A pliant method for anisotropic mesh generation. In: 5th Intl. Meshing Roundtable. (1996) 63–74
18. Yamakawa, S., Shimada, K.: High quality anisotropic tetrahedral mesh generation via ellipsoidal bubble packing. In: IMR. (2000) 263–274
19. Labelle, F., Shewchuk, J.R.: Anisotropic voronoi diagrams and guaranteed-quality anisotropic mesh generation. In: Proc. of SCG '03, New York, ACM Press (2003) 191–200
20. Boissonnat, J.D., Wormser, C., Yvinec, M.: Anisotropic diagrams: Labelle shewchuk approach revisited. In: CCCG. (2005) 266–269
21. Yokosuka, Y., Imai, K.: Guaranteed-quality anisotropic mesh generation for domains with curves. In: Proc. of EWCG'06. (2006)

22. Feng, Z., Hotz, I., Hamann, B., Joy, K.I.: Anisotropic noise samples. IEEE Transactions on Visualization and Computer Graphics **14** (2008) 342–354
23. Du, Q., Wang, D.: Anisotropic centroidal Voronoi tessellations and their applications. SIAM Journal on Scientific Computing **26** (2005) 737–761
24. Valette, S., Chassery, J., Prost, R.: Generic remeshing of 3d triangular meshes with metric-dependent discrete Voronoi diagrams. IEEE Trans Visu Comp Graph **14** (2008) 369–381
25. Peyré, G., Cohen, L.D.: Geodesic remeshing using front propagation. Int. J. Comput. Vision **69** (2006) 145–156