

# Image Compression with Anisotropic Geodesic Triangulations

Sébastien Bougleux  
GREYC  
Université de Caen

sebastien.bougleux@unicaen.fr

Gabriel Peyré    Laurent Cohen  
CEREMADE  
Université Paris Dauphine

{peyre,cohen}@ceremade.dauphine.fr

## Abstract

*We propose a new image compression method based on geodesic Delaunay triangulations. Triangulations are generated by a progressive geodesic meshing algorithm which exploits the anisotropy of images through a farthest point sampling strategy. This seeding is performed according to anisotropic geodesic distances which force the anisotropic Delaunay triangles to follow the geometry of the image. Geodesic computations are performed using a Riemannian Fast Marching, which recursively updates the geodesic distance to the seed points. A linear spline approximation on this triangulation allows to approximate faithfully sharp edges and directional features in images. The compression is achieved by coding both the coefficients of the spline approximation and the deviation of the geodesic triangulation from an Euclidean Delaunay triangulation. Numerical results show that taking into account the anisotropy improves the approximation by isotropic triangulations of complex images. The resulting geodesic encoder competes well with wavelet-based encoder such as JPEG-2000 on geometric images.*

## 1. Introduction

Image compression requires the extraction of meaningful features of an image to minimize the rate distortion curve. A popular tool is the wavelet transform, that offers a sparse representation of natural images. Wavelet coefficients are mapped to a binary code using an adaptive arithmetic encoder such as in the JPEG-2000 standard, see for instance [13]. Because of their square support, wavelets are however sub-optimal to extract geometric features such as edges or directional textures.

**Geometric compression.** Anisotropic representations such as curvelets [4] or bandlets [16] allows one to improve over the wavelet approximation, but their extension to practical coders remains difficult. Another attractive representation is the design of an adaptive triangulation to approximate the image with linear splines whose supports follow the geo-

metric features of the image. Optimal adaptation of triangulations is a NP-hard problem [1], and one has to use approximate greedy schemes to design the sampling and the triangulation layout.

**Triangulation simplification.** Methods from computer graphics based on mesh simplification are efficient for surface approximation [7]. Similar methods based on thinning together with Euclidean Delaunay triangulations have been applied to image compression [5]. Using an Euclidean metric might be problematic in highly anisotropic regions where an optimal triangulation deviates from an Euclidean Delaunay triangulation.

**Triangulation refinement.** Mesh adaptation using vertex insertion is popular for the resolution of PDEs whose solutions might exhibit singular features [8]. The first attempt to use anisotropic metric for Delaunay refinement [11] makes use of an approximate Riemannian geodesic metric.

**Contributions.** We propose to define a truly Riemannian geodesic sampling, and use the anisotropic Delaunay triangulation for image approximation. This combines in a common framework several important features from previous works, including fast Delaunay insertions, anisotropic sampling [6, 15], anisotropic triangulations [12, 11, 17, 2, 3]. We show that these properties are indeed crucial to improve over the state of the art for the compression of geometric images.

## 2. Approximation by Triangulation and Anisotropy

### 2.1. Image approximation by triangulations

It is possible to approximate an image, expressed as a function  $f \in L^2(\Omega)$ , using  $m$  linear spline functions defined on a triangulation  $(\mathcal{V}, \mathcal{F})$ , where  $\mathcal{V} = \{v_i\}_{i=1}^n \subset \Omega$  denotes the vertex set and  $\mathcal{F} \subset \{1, \dots, m\}^3$  the triangle set. Each triangle of  $\mathcal{F}$  is represented as a triplet of vertex indices. The resulting approximated image is given by

$$f_m = \sum_i a_i \varphi_i, \quad (1)$$

where  $\varphi_i$  is linear on each triangle of  $\mathcal{F}$ , and  $\varphi_i(v_j) = 1$  if  $i = j$  and 0 otherwise. The coefficients  $a_i$  are computed such that  $f_m$  is the orthogonal projection of  $f$  on the space  $V$  generated by the  $\varphi_i, e.g.$

$$f_m = \operatorname{argmin}_{g \in V} \|f - g\|. \quad (2)$$

This leads to solve the linear system  $\Phi a = F$ , where  $\Phi_{i,j} = \langle \varphi_i, \varphi_j \rangle$  and  $F_i = \langle f, \varphi_i \rangle$ .

To minimize the error  $\|f - f_m\|$  relative to the triangulation  $(\mathcal{V}, \mathcal{F})$ , it is necessary to optimize the position of the vertices and the connections between these vertices. We must put more points near areas of strong image gradient, and use strongly anisotropic triangles oriented in the direction of contours. For a cartoon image, which is a  $C^2$  function outside  $C^2$  contours, such an optimization of the triangulation leads to an error in  $\|f - f_m\| = O(m^{-1})$ , see for instance [13, 4]. In this case, the side length  $\Delta_1$  and  $\Delta_2$  of the triangles must verify  $\Delta_2 \approx \Delta_1^2$ . We propose an efficient triangulation algorithm adapted to this problem. This latter result is far better than the wavelet  $m$ -terms approximation, based on isotropic triangulations, which generates an error in  $O(m^{-1/2})$ .

## 2.2. Anisotropic metric

The desired anisotropy of a triangulation  $(\mathcal{V}, \mathcal{F})$  can be represented by a tensor field  $H : \Omega \rightarrow \mathbb{R}^{2 \times 2}$ . At each point  $x \in \Omega$ , the local metric  $H(x)$  characterizes the shape (its size and its orientation) of the mesh elements surrounding  $x$ . It can be defined as a symmetric positive definite matrix, and diagonalized as

$$H(x) = \lambda_1(x)e_1(x)e_1(x)^T + \lambda_2(x)e_2(x)e_2(x)^T, \quad (3)$$

where the eigenvalue fields  $\lambda_1$  and  $\lambda_2$  satisfying  $0 < \lambda_1(x) \leq \lambda_2(x)$ , and  $e_1, e_2$  are the orthogonal unoriented eigenvector fields.

Each triangle of  $(\mathcal{V}, \mathcal{F})$  conforming to  $H$  must be inscribed in an ellipsoid centered at  $x_0$

$$(x_0 - x)^T H(x_0)(x_0 - x) \leq t_m,$$

where  $t_m$  depends on the number  $m$  of triangles in  $\mathcal{F}$ . Triangle sides are thus aligned with  $e_1(x_0)$ , and they satisfy the ratio  $\Delta_1/\Delta_2 \approx \sqrt{\lambda_2(x_0)/\lambda_1(x_0)}$ .

For a  $C^2$  function  $f$ , the ratio  $\lambda_2(x_0)/\lambda_1(x_0)$  can be taken as the ratio  $|\mu_2(x_0)/\mu_1(x_0)|$  of the eigenvalues of the Hessian of  $f$  at  $x_0$ . This improves the constant in the approximation error  $\|f - f_m\| = O(m^{-1})$  [14]. However, this estimate is quite unstable and cannot be used to treat noisy images, oriented textures, or cartoon images with non-smoothed contours.

So, we propose to use the structure tensor [10] which is computed from the image gradient by

$$T_\sigma(x) = G_\sigma * T_0(x), \quad (4)$$

where  $G_\sigma$  is a Gaussian kernel applied to each component of the 1-rank tensor  $T_0(x) = (\nabla_x f)^T \nabla_x f$ . The parameter  $\sigma$  allows a robust estimation in the presence of noise. We have chosen  $\sigma = 5/n$  for an image of size  $n \times n$ .

From the diagonalization  $T(x) = \mu_1(x)e_1(x)e_1(x)^T + \mu_2(x)e_2(x)e_2(x)^T$ , we define the triangulation tensor as

$$H(x) = (\epsilon + \mu_1(x))^\alpha e_1(x) + (\epsilon + \mu_2(x))^\alpha e_2(x), \quad (5)$$

where  $\epsilon > 0$  controls the isotropic adaptivity of the triangulation (the variation of density, see [17]), and  $\alpha > 0$  controls the anisotropic adaptivity. A metric with a high value of  $\epsilon$  is almost constant, and a metric with a low value of  $\alpha$  is almost isotropic. We have fixed  $\epsilon$  to  $10^{-5}$  and the influence of the parameter  $\alpha$  is adapted to the type of images to process.

## 2.3. Geodesic distance

As explained in Section 2.2, at each point  $x_0 \in \Omega$ , the local metric  $H(x_0)$  defines the size of triangles around  $x_0$ , which must be of constant size  $t_m$  for the modified metric

$$\|x_0 - x\|_{H(x_0)} = \sqrt{(x_0 - x)^T H(x_0)(x_0 - x)}.$$

We compute a point set that achieves this condition by sampling them uniformly according to a global metric  $d_H(x, y)$  between any pair of points  $(x, y) \in \Omega \times \Omega$ .

The length of a piecewise smooth curve  $\gamma : [0, 1] \rightarrow \Omega$ , according to the metric  $H$ , is measured by

$$\begin{aligned} L_H(\gamma) &= \int_0^1 \|\gamma'(t)\|_{H(\gamma(t))} dt, \\ &= \int_0^1 \sqrt{\gamma'(t)^T H(\gamma(t)) \gamma'(t)} dt. \end{aligned}$$

A curve  $\gamma$  reaching a point  $\gamma(t) = x$  of  $\Omega$ , with speed  $\gamma'(t)$ , has a shorter local length if  $\gamma'(t)$  is approximately collinear to  $e_1(x)$  rather than to another direction. The geodesic distance between two points  $x$  and  $y$  of  $\Omega$  is the length of the shortest curve joining  $x$  and  $y$

$$d_H(x, y) = \min_{\gamma \in P(x, y)} L(\gamma), \quad (6)$$

where  $P(x, y) = \{\gamma : \gamma(0) = x \wedge \gamma(1) = y\}$  denotes the set of piecewise smooth curves joining the points  $x$  and  $y$ . A curve  $\gamma$ , which satisfies  $L_H(\gamma) = d_H(x, y)$ , is a geodesic.

Given a point set  $S = \{x_k\}_k \subset \Omega$ , its minimal action map is defined as

$$U_S(x) = \min_{x_k \in S} d_H(x, x_k). \quad (7)$$

An important theoretical result is if  $H$  is a smooth field,  $U_S$  is the unique viscosity solution of the following Hamilton-Jacobi equation

$$\|\nabla_x U_S\|_{H(x)^{-1}} = 1, \quad \text{with } U_S(x_k) = 0, \quad \forall k. \quad (8)$$

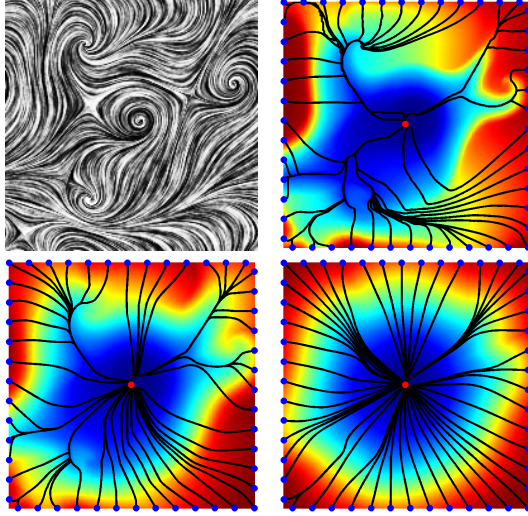


Figure 1. Examples of geodesic distance map  $U_S$  from a central point  $S = \{x_0\}$ , together with a few geodesic curves. The metric  $H$  is computed from the image  $f$  on the left using the structure tensor (5) with an increasing value of  $\epsilon$  and  $\alpha = .75$ , so that the metric becomes more isotropic from left to right.

The numerical solution of this equation can be computed on a regular grid of  $N = n \times n$  pixels. In the case of an isotropic metric  $H(x) = \lambda(x)Id_2$ , the Fast Marching (FM) algorithm, introduced by Sethian [19], allows to efficiently solve a discretization of (8) in  $O(N \log N)$  operations. Several extensions of the FM algorithm have been proposed to solve (8) for a generic metric, see for instance [18]. Figure 1 shows examples of geodesics distance and minimal paths.

### 3. Anisotropic Geodesic Meshes

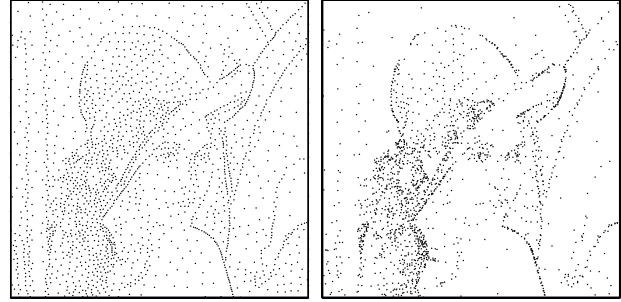
Computing a triangulation  $(\mathcal{V}, \mathcal{F})$  conformed to a tensor field  $H$  is a difficult optimization problem. The conformity constraint on  $\mathcal{V}$  imposes that the vertices must be uniformly distributed according to the geodesic distance  $d_H$ . This implies that  $d_H(v_i, v_j)$  is approximately constant if  $(v_i, v_j)$  is an edge of a face of  $\mathcal{F}$ . A uniform distribution of the vertices satisfies

$$\forall x \in \Omega, U_{\mathcal{V}}(x) \leq \delta, \quad (9)$$

where  $\delta > 0$  controls the density of the vertex set. Such a sample set of  $\Omega$  can be computed using a farthest point sampling strategy.

#### 3.1. Anisotropic farthest point sampling

The farthest point strategy (FPS) has been introduced by Eldar *et al.* [6] for image sampling. It is an iterative method which starts with an initial sample set  $\mathcal{V} \subset \Omega$ . Each iteration places a new sample at the farthest point  $p \in \Omega$  from the



(a) isotropic (b) anisotropic

Figure 2. Results of the FM-FPS algorithm for 2000 samples.

previously computed samples

$$p = \operatorname{argmax}_{q \in \Omega} U_{\mathcal{V}}(q), \quad (10)$$

until condition (9) is reached. This sampling strategy is summed up by the following algorithm:

**Algorithm:** FPS( $\mathcal{V}, \epsilon$ )

1. Find the point  $p \in \Omega$  satisfying (10).
2. If  $U_{\mathcal{V}}(p) < \epsilon$ , then exit, otherwise set  $\mathcal{V} \leftarrow \mathcal{V} \cup \{p\}$ .
3. Update the distance map  $U_{\mathcal{V}}$ , and goto 1.

Step 3 only requires a local update of the distance map in the region  $\{x \setminus U_{\{q\}}(x) \leq U_{\mathcal{V} - \{q\}}(x)\}$ . This makes the whole seeding process becoming faster as new points are inserted.

In the case of the Euclidean distance, the FPS is linked to the incremental construction of Voronoi diagrams. As pointed out by Eldar *et al.* [6], the farthest point from the set  $\mathcal{V}$  corresponds to the farthest vertex of the Voronoi diagram of  $\mathcal{V}$ , which is to the center of the largest ball not containing any point of  $\mathcal{V}$ . This nice property allows to compute efficiently the farthest point from the Voronoi diagram. However, it does not extend to the case of geodesic distances, for which the farthest point does not necessarily correspond to a vertex of the geodesic Voronoi diagram. Even if this drawback is not taken into account, computing geodesic Voronoi diagrams is not possible with classical computational geometry techniques (see for instance [12, 11]).

To sample images with the FPS, Eldar *et al.* [6] proposed to assign a weight to the vertices of the (Euclidean) Voronoi diagram with an application-dependent function, and to select the next point to include in  $\mathcal{V}$  as the Voronoi vertex having the maximum weight. Later, Peyré and Cohen [17], and Moening and Dodgson [15], proposed independently to approximate the geodesic distance with the fast marching method, leading to a fast marching FPS (FM-FPS).

With an anisotropic fast marching algorithm, the proposed extension of the FM-FPS to anisotropic metrics is obvious. Nevertheless, it should be useful for the resolution of problems which need a step of point selection.

Using the structure tensor (4) as a metric, the anisotropic FM-FPS concentrates the samples near the most important image structures, see Figure 2 for a comparison with the isotropic FM-FPS (computed with  $H(x) = \mu_1(x)\text{Id}_2$ ).

In the sequel, we present a modified version of the FPS algorithm which generates triangulations of images conformed to a metric  $H$ .

### 3.2. Anisotropic farthest point meshing

We have modified the farthest point strategy so that the faces of the triangulation  $(\mathcal{V}, \mathcal{F})$  are aligned with the direction of anisotropy  $e_1$ . This is imposed by defining  $\mathcal{F}$  from the dual complex of the anisotropic Voronoi diagram of the sample set  $\mathcal{V}$ .

**Anisotropic Voronoi diagram.** From the anisotropic geodesic distance  $d_H$ , the anisotropic Voronoi cell  $V_i(\mathcal{V})$  generated by a point  $x_i \in \mathcal{V}$  is defined as the set of points  $p \in \Omega$  for which the geodesic between  $p$  and  $x_i$  is shorter than the geodesics between  $p$  and the other points of  $\mathcal{V}$

$$V_i(\mathcal{V}) = \{x \in \Omega : d_H(x, x_i) \leq d_H(x, x_j), \forall x_j \in \mathcal{V}\}.$$

The union of all Voronoi cells, for  $i = 1, \dots, n$ , partition  $\Omega$  to form the bounded anisotropic Voronoi diagram of  $\mathcal{V}$ , noted  $V(\mathcal{V})$ . This diagram is also known as the Riemannian Voronoi diagram [12].

**Anisotropic Delaunay complex.** The dual complex of  $V(\mathcal{V})$  is the anisotropic Delaunay triangulation of  $(\mathcal{V}, \text{noted } D(\mathcal{V}))$ . Contrary to the (Euclidean) Delaunay triangulation, its anisotropic version  $D(\mathcal{V})$  does not necessarily define a valid planar triangulation:

- Some vertices can be isolated (connected to only one other vertex of  $D(\mathcal{V})$ ).
- Some edges can overlap.

Also, when  $\mathcal{V}$  is the result of the FPS,  $D(\mathcal{V})$  is not necessarily a triangulation of  $\Omega$ . Indeed, the boundary  $\partial\Omega$  should always be represented in  $D(\mathcal{V})$ .

**Farthest point meshing.** To construct a valid planar triangulation of  $\Omega$  according to the density constraint (9), we initialize the set  $\mathcal{V}$  with at least the four corner points of  $\partial\Omega$ , such that the boundary of  $D(\mathcal{V})$  corresponds to  $\partial\Omega$ . Then the FPS is modified as follows.

The farthest point  $x_k$  (see (10)) is inserted in  $\mathcal{V}$  if  $V_k(\mathcal{V}) \cap \partial\Omega = \emptyset$ . Otherwise, there exists a boundary edge  $(x_i, x_j) \in D(\mathcal{V})$  which satisfies  $V_k(\mathcal{V}) \cap (x_i, x_j) \neq \emptyset$ . In this case, the edge is said to be encroached by  $x_k$ . To ensure that  $\partial\Omega$  is always represented in  $D(\mathcal{V})$ , the encroached edge is split into two sub-edges  $(x_i, \tilde{x}_k)$  and  $(\tilde{x}_k, x_j)$ , where  $\tilde{x}_k = \operatorname{argmin}_{x \in (x_i, x_j) \cap V_k(\mathcal{V})} d_H(x, x_k)$ .

Once the FPS with boundary constraints has been computed, the proposed algorithm iteratively inserts points to  $\mathcal{V}$  while the triangulation is not planar:

- If a vertex  $x_i \in D(\mathcal{V})$  is isolated, it is connected to one vertex  $x_j \in D(\mathcal{V})$ . In this case, the point  $\hat{x}_i = \operatorname{argmin}_{x \in V_i(\mathcal{V}) \cap V_j(\mathcal{V})} d_H(x, x_i)$  is inserted to  $\mathcal{V}$ .
- If a triangle  $(x_i, x_j, x_l)$  is inverted, its dual Voronoi vertex  $V_{i,j,l} = V_i(\mathcal{V}) \cap V_j(\mathcal{V}) \cap V_l(\mathcal{V})$  is inserted to  $\mathcal{V}$ . When there is no inverted triangle, then no edge can overlap.

The farthest point meshing method is summed up by the following algorithm:

**Algorithm:** FPM( $\mathcal{V}, \epsilon$ )

1. Find the point  $x_k \in \Omega$  satisfying (10).
2. If  $U_{\mathcal{V}}(x_k) < \epsilon$  goto 5.
3. If  $x_k$  encroaches a boundary edge  $(x_i, x_j) \in D(\mathcal{V})$   
Split  $(x_i, x_j)$  into  $(x_i, \tilde{x}_k)$  and  $(\tilde{x}_k, x_j)$   
 $\mathcal{V} \leftarrow \mathcal{V} \cup \{\tilde{x}_k\}$ , update  $U_{\mathcal{V}}$  and goto 1.
4.  $\mathcal{V} \leftarrow \mathcal{V} \cup \{x_k\}$ , update  $U_{\mathcal{V}}$  and goto 1.
5. If  $\exists x_i \in D(\mathcal{V})$  isolated, then  $x_k \leftarrow \hat{x}_i$  and goto 3.
6. If  $\exists$  inverted triangle  $(x_i, x_j, x_l) \in D(\mathcal{V})$   
 $x_k \leftarrow V_{i,j,l}$  and goto 3.

This algorithm is similar to the anisotropic Delaunay refinement meshing algorithm of Labelle and Shewchuk [11, 2]. The main difference is we use a truly anisotropic geodesic distance, meanwhile they use the modified metric (3). This is also the strategy adopted by Bougleux *et al.* [3], but in a different context. This allows to obtain better results, in particular for triangulations with a few number of vertices, where the approximation with (3) can be important.

## 4. Image Approximation and Compression

### 4.1. Spline approximation

Once the image  $f$  has been triangulated with the anisotropic farthest point meshing method, its spline approximation  $f_m$ , given by (1), is progressively refined as the value of  $m$  increases. Figure 5 (left column) shows examples of triangulations for two values of  $m$  and  $\alpha = 0.75$ .

As shown in Figure 4, we found the error  $\|f - f_m\|$  minimal for  $\alpha \approx 0.75$ , with  $f_m$  the image of Figure 5 (bottom left), and this for a wide range of  $m$ . Following an experimental study,  $\alpha \approx 0.75$  seems to be an acceptable choice for a wide range of natural images. From now on, that is the value we are going to use in the paper.

Figure 3 shows the evolution of SNR, according to  $m$ , for an anisotropic Delaunay triangulation. It is compared to the SNR of the Euclidean Delaunay triangulation of the sample

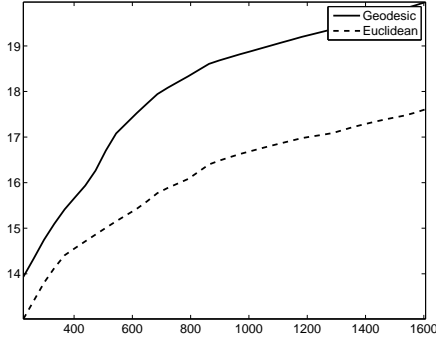


Figure 3. For  $\alpha = 0.75$ , SNR according to  $m$  for the anisotropic Delaunay triangulation (solid) and for the Euclidean Delaunay triangulation (dotted).

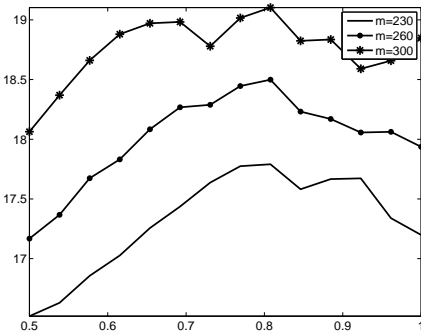


Figure 4. For  $m = 230, 260, 300$ , SNR according to  $\alpha$  (bottom).

set generated with  $\alpha = 0.75$ . This shows the importance to compute both an appropriate sampling and a suitable connectivity to obtain a better approximation.

## 4.2. Quantification

To encode the approximation  $f_m$ , we quantify the position of the vertices of  $\mathcal{V}$  with a setp  $T_v$ , as well as the coefficients  $a_i$  with a step  $T_a$ . Thus, we obtain rounded integer values  $\bar{v}_i = [a_i / T_v]$  and  $\bar{a}_i = [a_i / T_a]$ . The coefficients  $\bar{a}$  and  $\bar{v}$  are encoded with an arithmetic encoder on  $R_v + R_a$  bits. The decoder can compute the decompressed positions  $\tilde{\mathcal{V}} = \{\tilde{v}_i\}_i$  and coefficients by  $\tilde{v}_i = (\bar{v}_i + \text{sign}(\bar{v}_i) / 2)T_v$  and  $\tilde{a}_i = (\bar{a}_i + \text{sign}(\bar{a}_i) / 2)T_a$ . The decompressed approximation is thus given by

$$\tilde{f}_m = \sum_i \tilde{a}_i \tilde{\varphi}_i,$$

where  $\tilde{\varphi}_i$  is the spline basis associated to the vertex  $\tilde{v}_i$ .

The triangulation connectivity must also be encoded, on  $R_c$  bits. In order to reduce this number  $R_c$ , we transform the anisotropic Delaunay triangulation  $(\tilde{\mathcal{V}}, \mathcal{F})$  to the Euclidean Delaunay triangulation of  $\tilde{\mathcal{V}}$ , which can be constructed by the encoder and the decoder. We use the local optimiza-

tion procedure of Lawson (see for instance [9]) to compute this transformation. Given an interior edge  $(v_i, v_j) \subset \bar{F}$  incident to the triangles  $(v_i, v_j, v_k)$  and  $(v_i, v_j, v_l)$ , this edge is replaced by the edge  $(v_k, v_l)$  if the triangles are not in the Euclidean Delaunay triangulation. The number of bits needed to encode  $\mathcal{F}$  is reduced to  $R_c = K \log_2(A)$ , where  $K$  is the number of substitutions and  $A$  the number of edges in  $\mathcal{F}$ .

Other image compression methods, based on triangulations, have already been proposed in litterature, see for instance [5]. These methods begin with a dense triangulation which is decimated to reduce the approximation error. Our approach is different, it progressively refines the triangulation, which is very fast for low values of  $m$ . Moreover, it uses a Delaunay triangulation based on an anisotropic geodesic distance, where the elongation of triangles easily adapts to the image structures, contrary to the Euclidean Delaunay triangulation used for instance in [5].

The choice of  $T_v$  and  $T_a$  can be optimized to minimize the curve of coding rate  $\|f - f_m\|$  according to the number of bits  $R = R_v + R_a + R_c$ . To simplify matters, we have used  $R_v = 10^{-2}$  and  $R_a = \|f\|_\infty / 100$ . Figure 5 (left) compares compressions given by our method and JPEG-2000 (see [13]). This shows that for low coding rate, and for images with sufficient geometric structures, our method improves the current standard.

## 5. Conclusion and Perspectives

We have presented a new image compression method based on anisotropic geodesic Delaunay triangulations. The anisotropy of images is exploited to generate triangulations through a farthest point sampling approach. Images are then approximated by linear splines on the triangulations. Finally, the compression is achieved by coding both the coefficients of the spline approximation and the deviation between the geodesic Delaunay triangulation and the Euclidean one. By experimentation, we have shown that the resulting geodesic encoder competes well with wavelet-based encoder such as JPEG-2000.

Since anisotropic geodesic distances can also be computed on meshes with fast marching methods, the proposed approach can be extended, with some modifications, to compress meshes.

## References

- [1] P. Agarwal and S. Suri. Surface Approximation and Geometric Partitions. *SIAM J. Comput.*, 19:1016–1035, 1998. 1
- [2] J.-D. Boissonnat, C. Wormser, and M. Yvinec. Anisotropic diagrams: Labelle shewchuk approach revisited. In *CCCG*, pages 266–269, 2005. 1, 4
- [3] S. Bogleux, G. Peyré, and L. D. Cohen. Anisotropic geodesics for perceptual grouping and domain meshing. In

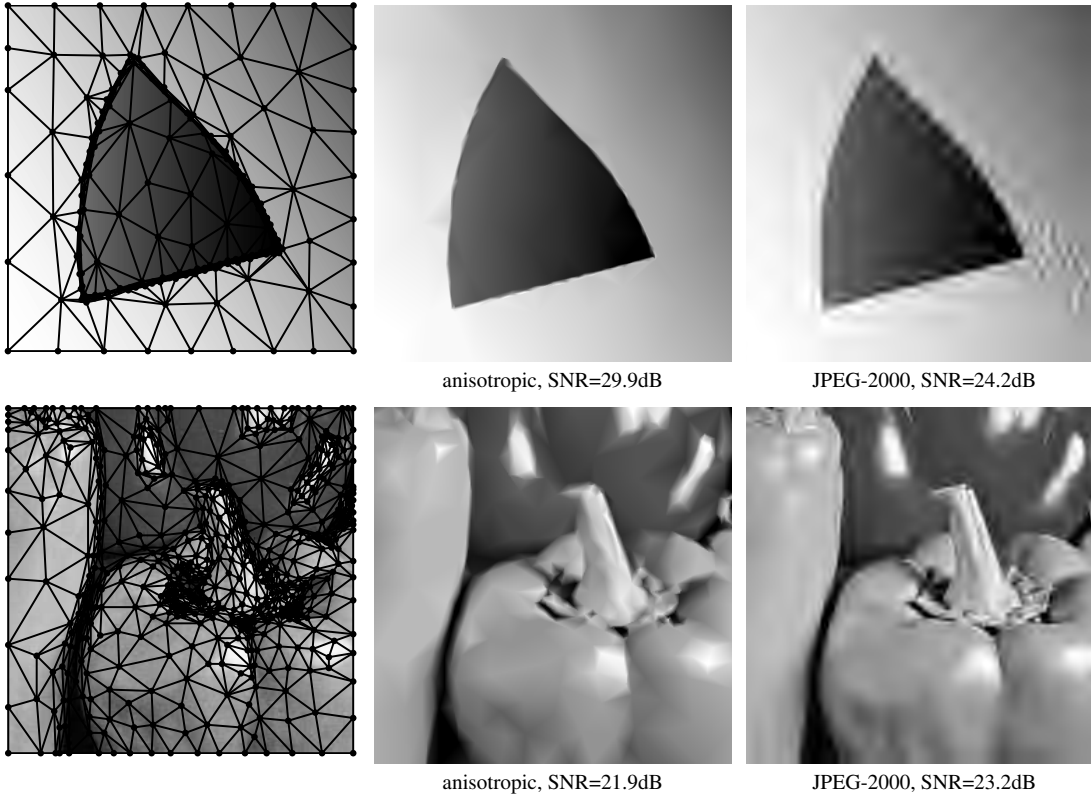


Figure 5. Comparison of our compression scheme with JPEG-2000, for the same number of bits, for  $m = 200$  (top) and  $m = 600$  (bottom) triangles, for  $\alpha = 0.75$ .

- ECCV '08: Proc. of the 10th European Conference on Computer Vision*, volume II, pages 129–142. Springer-Verlag, 2008. [1](#), [4](#)
- [4] E. Candès and D. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise  $C^2$  singularities. *Commun. on Pure and Appl. Math.*, 57(2):219–266, 2004. [1](#), [2](#)
- [5] L. Demaret, N. Dyn, and A. Iske. Image compression by linear splines over adaptive triangulations. *Signal Processing Journal*, 86(7):1604–1616, 2006. [1](#), [5](#)
- [6] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi. The farthest point strategy for progressive image sampling. *IEEE Trans. Image Processing*, 6(9):1305–1315, 1997. [1](#), [3](#)
- [7] M. Garland and P. Heckbert. Surface simplification using quadric error metrics. *Proc. of SIGGRAPH 1997*, pages 209–215, 1997. [1](#)
- [8] P. George, H. Borouchaki, P. Frey, P. Laug, and E. Saltel. Mesh generation and mesh adaptivity: theory, techniques. *Encyclopedia of computational mechanics*, E. Stein, R. de Borst and T.J.R. Hughes ed., John Wiley & Sons Ltd., 2004. [1](#)
- [9] O. Hjelle and M. Daehlen. *Triangulations and Applications*. Springer, 2006. [5](#)
- [10] M. Kass and A. Witkin. Analyzing oriented patterns. *Comput. Vision Graph. Image Process.*, 37(3):362–385, 1987. [2](#)
- [11] F. Labelle and J. R. Shewchuk. Anisotropic voronoi diagrams and guaranteed-quality anisotropic mesh generation. In *19th Annual Symp. on Computational Geometry*, pages 191–200. ACM, 2003. [1](#), [3](#), [4](#)
- [12] G. Leibon and D. Letscher. Delaunay triangulations and Voronoi diagrams for Riemannian manifolds. In *Proc. of the 16th Annual Symp. on Computational Geometry*, pages 341–349. ACM, 2000. [1](#), [3](#), [4](#)
- [13] S. Mallat. *A Wavelet Tour of Signal Processing*, 3<sup>rd</sup> edition. Elsevier, 2009. [1](#), [2](#), [5](#)
- [14] J.-M. Mirebeau and A. Cohen. Greedy bisection generates optimally adapted triangulations. Technical report, Laboratoire Jacques-Louis Lions, 2008. [2](#)
- [15] C. Moenning and N. A. Dodgson. Fast marching farthest point sampling. In *Eurographics 2003*, 2003. [1](#), [3](#)
- [16] E. L. Pennec and S. Mallat. Sparse geometric image representations with bandelets. *IEEE Trans. Image Proc.*, 14(4):423–438, Apr. 2005. [1](#)
- [17] G. Peyré and L. D. Cohen. Geodesic remeshing using front propagation. *Int. Journal on Computer Vision*, 69(1):145–156, 2006. [1](#), [2](#), [3](#)
- [18] E. Prados, C. Lenglet, J. P. Pons, N. Wotawa, R. Deriche, O. D. Faugeras, and S. Soatto. Control theory and fast marching methods for brain connectivity mapping. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 1076–1083. IEEE Computer Society, 2006. [3](#)
- [19] J. Sethian. *Level Sets Methods and Fast Marching Methods*. Cambridge University Press, 2nd edition, 1999. [3](#)