A Hierarchical Approach for Regular Centroidal Voronoi Tessellations

Li Wang, Franck Hétroy-Wheeler, Edmond Boyer

Volume xx (200y), Number z, pp. 1–15

A Hierarchical Approach for Regular Centroidal Voronoi Tessellations

L. Wang, F. Hétroy-Wheeler and E. Boyer

Univ. Grenoble Alpes & Inria & CNRS, LJK, F-38000 Grenoble, France

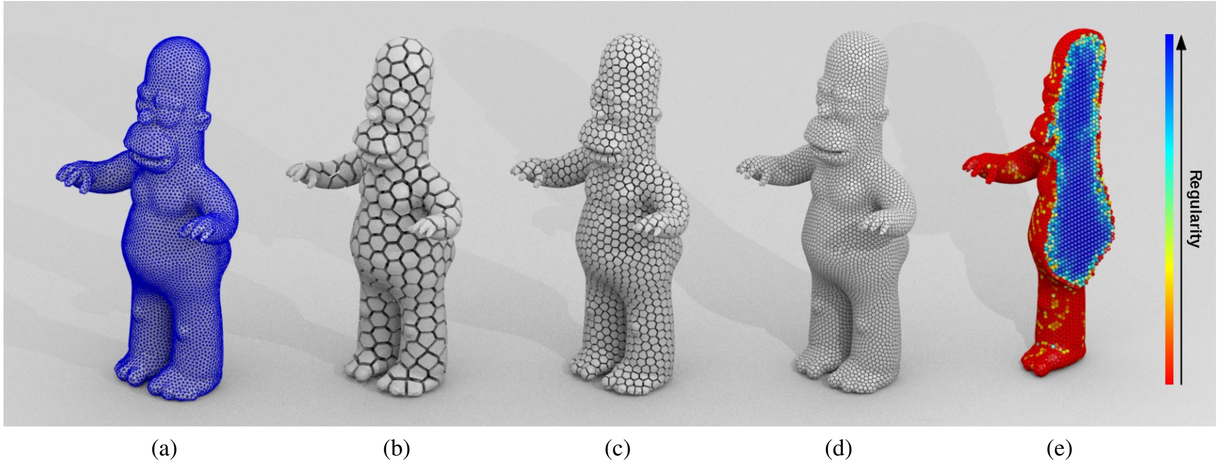


Figure 1: *Hierarchical CVT computation in 3D. (a) Input: a 3D object bounded by a triangulated mesh. (b,c,d) Successive CVTs computed using our approach, with* 546*,* 4375 *and* 35000 *cells respectively. (e) A cut of Homer shows that most of the interior Voronoi cells present high regularity values.*

Abstract

In this paper we consider Centroidal Voronoi Tessellations (CVTs) and study their regularity. CVTs are geometric structures that enable regular tessellations of geometric objects and are widely used in shape modeling and analysis. While several efficient iterative schemes, with defined local convergence properties, have been proposed to compute CVTs, little attention has been paid to the evaluation of the resulting cell decompositions. In this paper, we propose a regularity criterion that allows us to evaluate and compare CVTs independently of their sizes and of their cell numbers. This criterion allows us to compare CVTs on a common basis. It builds on earlier theoretical work showing that second moments of cells converge to a lower bound when optimising CVTs. In addition to proposing a regularity criterion, this paper also considers computational strategies to determine regular CVTs. We introduce a hierarchical framework that propagates regularity over decomposition levels and hence provides CVTs with provably better regularities than existing methods. We illustrate these principles with a wide range of experiments on synthetic and real models.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling —Curve, surface, solid, and object representations I.5.3 [Pattern Recognition]: Clustering—

1. Introduction

Centroidal Voronoi Tessellations (CVTs) are specific Voronoi diagrams in which each site is located at the centroid of the associated Voronoi cell. CVTs yield tessellations of 2D or 3D domains that have strong regularity properties. Consequently, they are widely used to represent shapes and structures in various scientific domains that include quantization, sensor networks, crystallography and shape modeling, among others (see e.g. [DFG99]). In this paper, we consider CVTs for 2D and 3D regions that are bounded by polygonal curves and 2-manifold meshes, respectively. Inspired by a recent work from Quinn et al. [QSL*12], we study CVT optimality with respect to the spatial regularity of the cells.

Over the last decades, even though many authors have considered the construction of CVTs, little effort has been devoted to the evaluation of their regularities. For many applications, such as climate modeling [JRG11] and shape tracking [AFB15], the regularity of the cell decomposition is crucial to ensure uniform local behaviour. Interestingly, for a few applications such as rendering and stippling, regularity should be avoided [BSD09]. Nonetheless, a measure is desirable to assess the regularity of a CVT. This is true also when comparing CVT decompositions of different shapes or of a single shape but with different decomposition levels. To the best of our knowledge, such a regularity measure has not yet been proposed for CVTs. In practice, they are usually evaluated using the CVT energy function (see Section 2.1), which integrates distances within cells. However, while this energy accounts for the compactness of the cells [LWL*09], it is a metric that depends both on the number of cells and on the volume of the shape.

In this paper, we build on the theoretical work of Conway and Sloane [CS82] and propose a regularity criterion based on the normalized second order moments of the cells. We show that this regularity criterion is linked to the CVT energy function but is dimensionless and therefore enables global evaluations as well as comparisons. We also consider computational strategies to build regular CVTs and we introduce a hierarchical approach that provides CVTs with more regularity than state-of-the-art methods. Our strategy is based on a subdivision scheme that preserves cell regularity and the (local) optimality of CVTs on unbounded domains. This scheme tends to propagate cell regularity through hierarchy levels when applied to bounded domains. We demonstrate the efficiency of this framework with an in-depth evaluation that includes sensitivity analysis, comparisons with previous work and analyses of the convergence speed and computation time.

The remainder of this paper is organized as follows. In Section 2, we review fundamental ideas and related work on CVTs. We introduce our CVT regularity criterion and detail our hierarchical framework in Sections 3 and 4, respectively. Section 5 evaluates the approach.

2. Background and Related Work

2.1. Centroidal Voronoi tessellation

Given a finite set of n points $X = \{x_i\}_{i=1}^{n}$, called *sites*, in a m-dimensional Euclidean space $\mathbb{E}^m$, the *Voronoi cell* or *Voronoi region* Ω_i [Aur91, For92, OBSC00] of x_i is defined as follows:

$$\Omega_i = \{x \in \mathbb{E}^m \mid \|x - x_i\| \leq \|x - x_j\|,\ \forall j \neq i\}.$$

The partition of $\mathbb{E}^m$ into Voronoi cells is called a *Voronoi tessellation*.

Some of the Voronoi cells are not closed. However, in many applications, only the intersection of the Voronoi cells with an input 2D or 3D object Ω are required. A *clipped Voronoi tessellation* [YWLL13] is the intersection between the Voronoi tessellation and the object. A *clipped Voronoi cell* is thus defined as:

$$\Omega_i = \{x \in \Omega \mid \|x - x_i\| \leq \|x - x_j\|,\ \forall j \neq i\}.$$

A *centroidal Voronoi tessellation* [DFG99] is a special type of clipped Voronoi tessellation where the site of each Voronoi cell is also its centre of mass. Let the clipped Voronoi cell Ω_i be endowed with a density function ρ such that $\rho(x) > 0\ \forall x \in \Omega_i$. The centre of mass $\hat{x}_i$, also called the centroid, of Ω_i is defined as follows:

$$\hat{x}_i = \frac{\int_{\Omega_i} \rho(x) x \, d\sigma}{\int_{\Omega_i} \rho(x) \, d\sigma},$$

where $d\sigma$ is the area differential.

CVTs are widely used to discretize 2D or 3D regions. In that respect, CVTs are optimal quantizers that minimise a distorsion or quantization error $E : \mathbb{E}^{nm} \to \mathbb{R}$ defined as:

$$E(X) = \sum_{i=1}^{n} F_i(X) = \sum_{i=1}^{n} \int_{\Omega_i} \rho(x) \|x - \hat{x}_i\|^2 \, d\sigma. \quad (1)$$

CVTs correspond to local minima of the above function E, also called CVT energy function [DFG99]. By definition, an *optimal CVT* achieves the global minimum of this function. Yet finding an optimal CVT appears difficult since the energy function is usually non linear and non convex [LWL*09, LSPW12]. The function E measures the quantization error of a Voronoi tessellation and expresses, to some extent, the compactness of the cells [LWL*09]. However, it does not quantify how regular a tessellation is since it depends on the dimensions of the original region as well as the number of cells considered. As stated in the introduction, our objective in this paper is the ability to quantify a CVT decomposition independently of the shape, size and cell number.

It should be noticed that although Delaunay tetrahedrizations are dual to the Voronoi tessellations, optimizing tetrahedral decompositions is a different problem without guaranties over the dual Voronoi tesselations. As pointed out by [ACSYD05], duals of CVTs are actually Delaunay tetrahedrizations that may contain badly shaped or irregular tetrahedra. In this paper we focus on CVT decompositions of volumetric shapes instead of Delaunay tetrahedrizations.

2.2. CVT methods

Existing CVT computation strategies are mostly iterative optimization that rely on two fundamental steps: (i) find the initial locations for the given number of sites; (ii) optimize the site locations by minimizing the CVT energy function E (Eq. (1)). They differ then by the initialization they consider and by the optimization approach they implement.

2.2.1. Initialisation

The initial position of the sites has a strong influence on the convergence speed and on the result quality. Different methods have been considered in the literature.

Random Sampling. The idea is to sample the initial site locations randomly inside the object. This simple and fast method is widely used. However, neither the speed of convergence nor the quality of the result can be guaranteed. Other sampling methods can be used to improve these criteria, such as farthest point sampling or Ward's method [MS06].

Greedy Edge-Collapsing. Moriguchi and Sugihara proposed a method which applies a greedy edge-collapsing decimation on the input object and uses the decimated mesh vertices as the initial site locations [MS06]. As pointed out by Quinn et al. [QSL*12], this method can be time-consuming, and the sites may not be regularly positioned if the object is not described by a regular mesh. Consequently, the quality of the resulting CVT can be even worse than using random sampling.

Hammersley Sampling. Quinn et al. suggest to use Hammersley sequences to generate the initial site locations [QSL*12]. Hammersley sequences have correlated positions, which means that the probability of a site being at some position depends on the positions of its neighbours. Unfortunately, the Hammersley sequence generation algorithm as described in [QSL*12] can only place the sites in a square in 2D or a cube in 3D. As a result, the number of sites in the region tessellation is difficult to control.

2.2.2. Iterative scheme

Most of the strategies update the site locations using the Lloyd's gradient descent method [Llo82]. At each iteration, this method moves the current sites to the centroid locations of the corresponding clipped Voronoi cells. This is the continuous equivalent to the k-means clustering algorithm in the discrete case. It has been proved that this leads the CVT energy function to reach a local minimum [DFG99]. Convergence speed can anyway be slow since the site locations may oscillate around a local minimum.

To speed up convergence, Du et al. proposed a Lloyd-Newton method [DE06] which is equivalent to minimizing the sum of distances between the sites and the centroids of the corresponding Voronoi cells. Unfortunately, the resulting tessellation is not always a proper CVT since it is not necessarily a local minimum of E. In an influential work, Liu et al. prove that CVT energy function has C^2 smoothness almost everywhere, except for some non-convex parts of the object [LWL*09]. According to this property, quasi-Newton methods can be used to minimize the CVT energy function. This leads to fast and effective updates in practice.

Another strategy worth mentioning here is the stochastic approach of Lu et al. [LSPW12]. In this iterative approach, the site locations are perturbed once a local mimimum of the energy function is reached and the algorithm is then launched again. The global minimum can theoretically be reached after an infinite number of iterations. In practice, convergence is still slow (see Section 5.3).

In this work, we focus on the regularity of CVTs rather than on the iterative scheme adopted to minimize the CVT energy function itself. The hierarchical algorithm we propose does not depend on this scheme but contributes with respect to the initialization step, in a way similar to the stochastic approach in [LSPW12]. In practice, we use a quasi-Newton approach in our implementation and in all the methods we used for comparisons because of its fast convergence.

3. Regularity Criterion

As mentioned before, the energy function E (Eq. (1)) measures the quantization error. It provides therefore a way to compare CVTs when the shape under consideration and the number of CVT sites are the same. However, this energy function does not evaluate the regularity of the cells and cannot be used for comparison when the number of cells, the shape or the size differ. In this section, we build on theoretical results on optimal quantizers to propose a measure for cell regularity.

3.1. Dimensionless second moment of a cell

In a seminal work, Gersho [Ger79] stated the conjecture that, for a sufficiently large number of sites, all cells of a distortion-minimising CVT are congruent to some polytope H, with the possible exception of regions touching the boundary of the tessellated object, where the polytope H only depends on the dimension m. Gersho's conjecture was proved in two dimensions [New82], the Voronoi cells being regular hexagons in that case. A weaker version of Gersho's conjecture was also proved in three dimensions [BS83]. It says that among all lattice-based CVTs (i.e., regular CVTs, where sites are located on a regular grid), the body-centered cubic (BCC) lattice is the optimal one. The BCC lattice has its sites displayed on a regular cubic grid, with additional sites at the centre of each cube. The Voronoi tessellation of a BCC lattice is called a bitruncated cubic honeycomb. Each of its cells is a truncated octahedra. Thus, Voronoi cells are truncated octahedra for optimal lattice-based CVTs in 3D. Our criterion is based on Gersho's conjecture and also on the following work of Conway and Sloane [CS82].

Motivated by the design of quantizers of m-dimensional Euclidean spaces, Conway and Sloane [CS82] define, for a given polytope $P \in \mathbb{E}^m$, the value:

$$G(P) = \frac{1}{m} \frac{\int_P \|x - \hat{x}\|^2 \, \mathrm{d}x}{\left(\int_P \mathrm{d}x\right)^{(m+2)/m}}, \tag{2}$$

with $\hat{x}$ the centroid of P.

$G(P)$ is called the *dimensionless second moment of* P. It is a measure that depends neither on the dimension m nor on the volume $\int_P \mathrm{d}x$ of P, only on its shape. In contrast, the CVT energy function E (Eq. (1)) reflects the average *unnormalised second moment* of Voronoi cells.

3.2. Regularity measure

Using Gersho's conjecture in the unbounded case, Conway and Sloane showed that the lower bound of $G(P)$ for any space-filling polytope in two dimensions is that of the hexagon:

$$G_2 = \frac{5}{36\sqrt{3}} = 0.0801875\ldots$$

Similarly, in three dimensions, and with unbounded lattices, the optimal lattice-based CVT being the Voronoi tessellation of a BCC lattice, the optimal quantizer is the truncated octahedron with the lower-bound G_3:

$$G_3 = \frac{19}{192\sqrt[3]{2}} = 0.0785433\ldots$$

Consequently, for a sufficiently large number of sites and with the exception of the boundary regions, an optimal CVT should present cells with values of G close to the optimal value G_m. Thus, G is a measure of the regularity of a CVT cell since in the limit, with an infinite number of sites, all cells should reach the value G_m. Note here that we assume a large number of cells and that this reasoning does not apply to the boundary cells, for which the optimal quantizers are not necessarily hexagons (truncated octahedra in 3D) nor necessarily space-filling polytopes. However, under the assumption that boundary cells are largely inferior to interior cells, the distribution of the values of G is a good indicator of the regularity of cells for a given CVT where the regularity is defined with respect to the dimensionless moment G. In our experiments, we consider the average value of G over a CVT:

$$\overline{G}(X) = \frac{1}{n} \sum_{i=1}^{n} G(\Omega_i). \tag{3}$$

3.3. Relation to the CVT energy

Under the assumption of a uniform density function ρ and using the definition (2), the CVT energy E (Eq. (1)) can be rewritten as:

$$E(X) = \sum_{i=1}^{n} m V(\Omega_i)^{(m+2)/m} G(\Omega_i),$$

where $V(\Omega_i) = \int_{\Omega_i} \mathrm{d}x$. Using Gersho's conjecture with unbounded lattices, optimal CVTs present in that case similar cells with volumes V/n and hence:

$$E(X) \sim mn \left(\frac{V}{n}\right)^{(m+2)/m} \overline{G}(X).$$

Thus optimizing the CVT energy E is equivalent to optimizing the average value $\overline{G}$ of G with infinite lattices. Knowing the theoretical optimal quantizer G_m in that case, we can even deduce that the value of an optimal CVT energy:

$$E_m = mn \left(\frac{V}{n}\right)^{(m+2)/m} G_m \tag{4}$$

These are theoretical values for infinite lattices. Nevertheless, with bounded shapes, our experiments show that optimal CVTs converge asymptotically to these values. Note anyway that although the CVT energy E and the regularity $\overline{G}$ are related by the above expression, optimizing $\overline{G}$ directly is inefficient since $\overline{G}$ is dimensionless and therefore ambiguous with respect to the cell sizes. The interest of $\overline{G}$ lies in the comparison between CVTs with different numbers of sites or of different volumes, which is not possible with E.

4. Hierarchical Centroidal Voronoi Tessellation

We now propose a new algorithm to compute CVTs that exploit regularity aspects. As shown by Lu et al. [LSPW12], computing a CVT with a small number of sites is more likely to be regular than with a large number of sites. Thus, we choose to follow a hierarchical approach, creating a coarse regular tessellation from a small number of sites and then refining it while preserving the regularity (see Figure 2).

Although local subdivision schemes have already been proposed for mesh generation (e.g. [TAD07, TWAD09] for 2D triangle and tetrahedral meshes, respectively), we do not know of a previous global hierarchical approach for CVT computation. In addition to the acceleration of the convergence speed, which is well known in many fields (a hierarchical sampling approach is for instance described for quantization in [GG91]), such an approach gives guarantees about the CVT regularity.

Our input is a 2D or 3D object, represented by its boundary: a closed polygonal curve in the first case and a manifold mesh without boundary in the second case. We also ask the user to provide the target number n of sites in the final CVT and the desired number s of subdivisions. From n and s

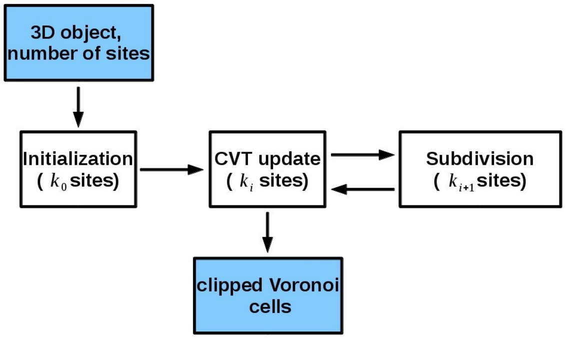

3D object, number of sites
Initialization (k_0 sites)
CVT update (k_i sites)
Subdivision (k_{i+1} sites)
clipped Voronoi cells

Figure 2: *Overview of our hierarchical approach.*

we derive an initial number k_0 of sites, as explained in Section 4.1. We elaborate on the choice of these parameters in Section 5.

We first create an initial CVT of the object with k_0 sites, using a standard algorithm, see Section 4.3. We then subdivide this tessellation, as explained in Section 4.1. This generates a new tessellation with k_1 sites, which is not Centroidal Voronoi. These sites are then moved towards the centroids of their cells to generate a new CVT with k_1 sites, as described in Section 4.2. We iterate the subdivision - CVT update process s times, until the desired number $k_s = n$ of sites has been reached.

We now detail the three stages of our approach by starting with the subdivision scheme.

4.1. Subdivision

The idea behind our subdivision scheme is to preserve the (local) optimality of the CVT. For example for the 2D case, a CVT is locally optimal with respect to our regularity criterion when its sites form an hexagonal lattice (see Figure 3 (a)), as explained in Section 3. Hence, our goal is to add sites such that the new set of sites keeps forming an hexagonal lattice. In this way, the new CVT will also be locally optimal with respect to regularity in the same area. In non-optimal areas, sites will move and possibly align to form a locally optimal lattice. Thus, iterating the subdivision - CVT update process tends to increase the area where the CVT is optimal for regularity, as shown on Figure 5. With a large number of subdivision s, most interior cells are expected to be regular.

Let X be a set of sites of a given CVT. To subdivide this CVT, we compute its dual Delaunay triangulation and add the centre of each Delaunay edge to X. As shown on Figures 3 and 4, this preserves the local optimality of the CVT.

4.1.1. Number of sites

The previous subdivision scheme does not account for the desired number n of sites. To set up the initial number of

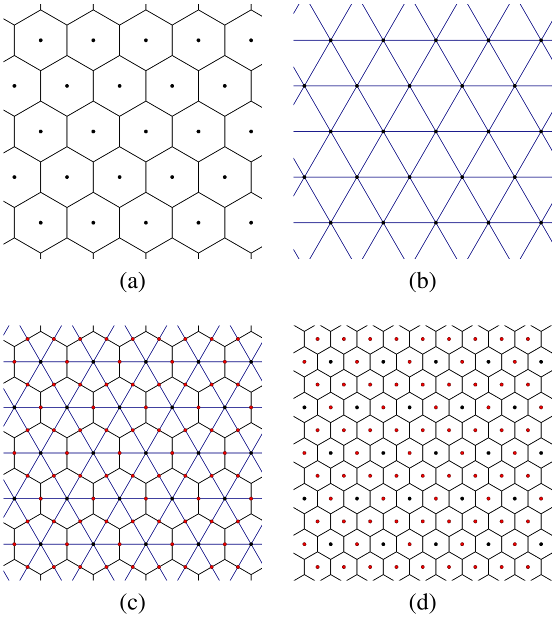


Figure 3: *Subdivision scheme (2D case). (a) Locally optimal CVT: the sites form an hexagonal lattice. (b) Delaunay triangulation of the sites. (c) Subdivision: sites are added in the centre of each edge of the Delaunay triangulation (in red). (d) The new set of sites also forms an hexagonal lattice.*

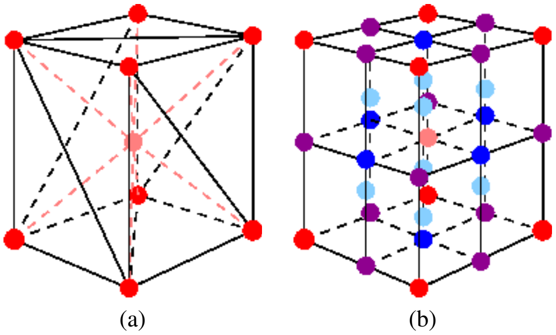


Figure 4: *Subdivision scheme (3D case). (a) Delaunay triangulation of the sites, which form a BCC lattice. (b) Subdivision: sites are added in the centre of each edge of the Delaunay triangulation (in blue and purple). The new set of sites also forms a BCC lattice.*

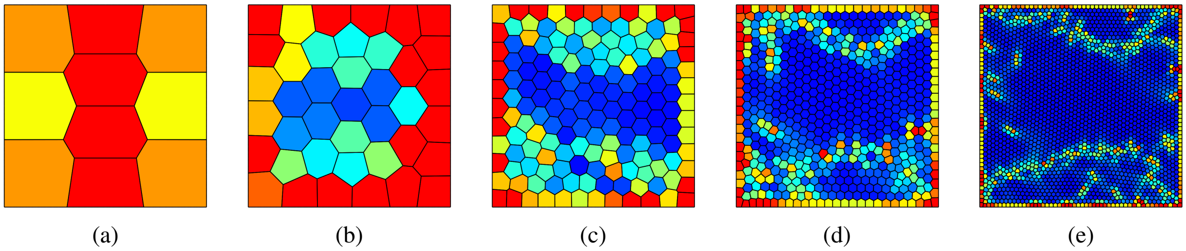


Figure 5: *Hierarchical CVT computation. From an initial CVT with* $k_0 = 10$ *sites (left), successive subdivisions and updates lead to CVTs with* $k_1 = 40$*,* $k_2 = 160$*,* $k_3 = 640$ *and* $k_4 = 2560$ *sites (from left to right). The cell regularity measure* $G_m(\Omega_i)$ *is colour-coded from blue (regular) to red (far from regular). Note how regular areas grow.*

sites k_0 such that it reaches the value n after s subdivisions, we proceed in the following way.

Let X be an hexagonal lattice, that is a to say an optimal 2D CVT, with k_i sites. Our subdivision scheme generates a new tessellation with $k_{i+1} = k_i + \frac{6k_i}{2} = 4k_i$ sites, since a new site is created on each of the six edges of a cell, an edge is shared by two cells and there are k_i cells. In the optimal 3D case (BCC lattice), the same reasoning shows that $k_{i+1} = k_i + \frac{14k_i}{2} = 8k_i$, since a site is added on each of the 14 faces of a truncated octahedron. The maximum number of iterations to reach n from a small number k_0 of sites in these ideal cases is thus $s = \lfloor \log_4(n) \rfloor$ and $\lfloor \log_8(n) \rfloor$, respectively.

Thus, if $s \geq \lfloor \log_a(n) \rfloor$, with $a = 4$ in the 2D case and $a = 8$ in the 3D case, we change s to $\lfloor \log_a(n) \rfloor$. We then define s numbers $b_1, \ldots, b_s$ such that $b_s = n$ and $b_i = \lfloor b_{i+1}/a \rfloor$, $1 \leq i \leq s-1$. b_i represents the target number of sites after i iterations. We also define $k_0 = \lfloor b_1/a \rfloor$. After the i-th subdivision, we check the new number k_i of sites. In case of an optimal CVT, $k_i = b_i$. Otherwise, $k_i \leq b_i$. If k_i is smaller than b_i, we randomly sample $b_i - k_i$ new sites inside the *boundary* Voronoi cells. When a new site is inserted into a boundary cell, this cell is then removed from the list of candidate boundary cells for next insertions. This way, the regularity is empirically preserved as much as possible since sites are inserted in different boundary cells, be avoiding many new sites to be neighbours to each other. We thus have b_i sites after the i-th iteration, which are as regularly sampled as possible. This will improve the speed of the CVT update computation, which we describe in the next section.

As an example, Table 1 gives the number k_i of sites obtained after each subdivision and the number $b_i - k_i$ of sites added in the boundary cells, for CVTs depicted in Figures 5, 8 and 9.

4.2. CVT update

Once a new set of sites is defined, any CVT computation method can be used to move these sites towards the centroid of their Voronoi cells. In practice, we use the L-BFGS quasi-Newton algorithm, since it is known to be one of the fastest methods [LWL*09]. As explained in the previous section, the sites where the previous CVT was optimal are not moved, thanks to our subdivision scheme. As a consequence, although the number of sites has increased, the CVT computation is very fast (see Section 5.5 for a discussion).

Once the sites are moved to their new positions and the tessellation is computed, we clip it to the boundary mesh. Our clipping algorithm, detailed below, guarantees that the boundary of the tessellation is a triangulated mesh.

4.2.1. Clipping algorithm

Computing a clipped Voronoi tessellation of an arbitrary 3D object, usually described by its meshed boundary surface, is not an easy problem. Yan et al. [YWLL13] have proposed an algorithm to compute clipped Voronoi diagrams of 3D objects described by tetrahedral meshes. This algorithm consists of two main steps: detection of boundary sites by computing surface restricted Voronoi diagram [ES94, YLL*09] and computation of the intersection between the Voronoi cells of boundary sites and the input tetrahedral mesh using Sutherland's clipping algorithm [SH74]. Recently, Lévy proposed another efficient method based on iterative convex clipping [LÍ4]. This method expresses the clipping problem as a 3D volume intersection problem but also requires a tetrahedral mesh as input. When the input 3D object is given as a closed triangle mesh, a 3D constraint Delaunay triangulation must be computed first [She98, She08]. This is a complex problem which has many degenerate cases and usually requires additional (Steiner) points to ensure the existence of a solution. The complexity highly depends on the quality of the input surface triangle mesh [Eri03]. Inspired by [ZBH11], we overcome this problem and propose an algorithm that exploits a 2D constrained Delaunay triangulation to determine triangles on the input mesh that intersect a given Voronoi cell, without the need of a tetrahedral mesh inside the shape.

Our algorithm first triangulates the polygonal boundaries of the Voronoi cells. In case of an infinite Voronoi cell, the infinite rays edging the cell are replaced by finite length segments, with a length greater than the diameter of the input

Object	n	s	k_0	k_1	$b_1 - k_1$	k_2	$b_2 - k_2$	k_3	$b_3 - k_3$	k_4	$b_4 - k_4$	k_5	$b_5 - k_5$
Figure 8	1033	4	4	5	11	49	15	233	25	995	38	/	/
Figure 5	2560	4	10	27	13	145	15	609	31	2535	25	/	/
Figure 9	10025	5	9	23	16	130	26	568	58	2373	133	9780	245

Table 1: *Number of sites after each subdivision, and number of sites randomly inserted in boundary cells.*

object. The boundaries of the cell, now finites, are then triangulated. Since the boundary of the 3D object is given as a triangulated mesh, the clipping problem now reduces to the computation of triangle-triangle intersections. Once such intersections have been found, we set them as constraints. Constraints are represented as line segments. The intersection I of two triangles is processed according to the following rules:

- Case 1: if I is a point, ignore it.
- Case 2: if I is a line segment, add it to the constraints.
- Case 3: if I is a triangle, add its three edges to the constraints.
- Case 4: otherwise, I is a polygon, construct segments using adjacent vertices of this polygon and add them to the constraints.

These cases are illustrated in Figure 6. The interior of each intersected triangle of either the cell boundary or the mesh is then robustly triangulated using a 2D constrained Delaunay triangulation.

Our clipping algorithm is summarized below (Algorithm 1). Figure 7 depicts its main steps.

```
Data: cell $C$, 3D object $O$ bounded by a triangulated
      mesh $M$
Result: clipped cell $C'$
$C_T$ := TriangulateBoundary($C$);
$I$ := Intersection($C_T$, $M$);
if $I$ not empty then
    $T$ := IntersectedTriangles($C_T$, $I$);
    for each triangle $t$ of $T$ do
        $T_1$ := ConstrainedDelaunay($t$, $I$);
        for each triangle $t_1$ of $T_1$ do
            if IsInside($t_1$,$O$) then
                Add($t_1$,$C'$);
            end
        end
    end
    $T$ := IntersectedTriangles($M$, $I$);
    for each triangle $t$ of $T$ do
        $T_2$ := ConstrainedDelaunay($t$, $I$);
        for each triangle $t_2$ of $T_2$ do
            if IsInside($t_2$,$C$) then
                Add($t_2$,$C'$);
            end
        end
    end
end
else
    $C'$ := $C$;
end
```

Algorithm 1: Clipping algorithm.

4.3. Initialisation

Before starting the subdivision - CVT update process, we create a first coarse CVT from the input object, with a number k_0 of sites. Our aim is to get an initial CVT with as-large-as-possible optimal areas, since our subdivision scheme can only make these areas grow, as explained in Section 4.1. We propose two different possible initialisations, each of them having different benefits.

A first straightforward idea to initialise the hierarchical CVT computation is to create a CVT using random sampling and a L-BFGS quasi-Newton algorithm to update the positions of the sites. This approach is fast and easy to implement. However, the constructed CVT with k_0 sites may be far from being regular.

Another idea to create a coarse but regular CVT is to sample the k_0 sites on a optimal lattice (hexagonal lattice in 2D and BCC lattice in 3D) which includes the input object. The density of the lattice should be chosen so that there are k_0 sites inside the object. As stated by [QSL*12], it is very hard to control the number of sites inside the object because the density depends on both the size and the shape of the input object. However, tuning the density of the lattice to reach the proper number of sites inside the object is easier with a small number k_0 of sites than with a large number n sites. When applicable, this leads to an optimal tessellation, except at the boundary of the object. In addition, most sites do not need to be moved to create a CVT. Thus, this approach is fast and generates regular CVT cells, except on the boundary of the object.

Both initialisation methods are evaluated in the next section. For the random initialisation, 10 runs are performed for each test, and the median value is taken as the result.

5. Evaluation

We now provide a thorough evaluation of our approach. We first analyse the effect of our hierarchical approach over the regularity of the generated CVT, by discussing the influence

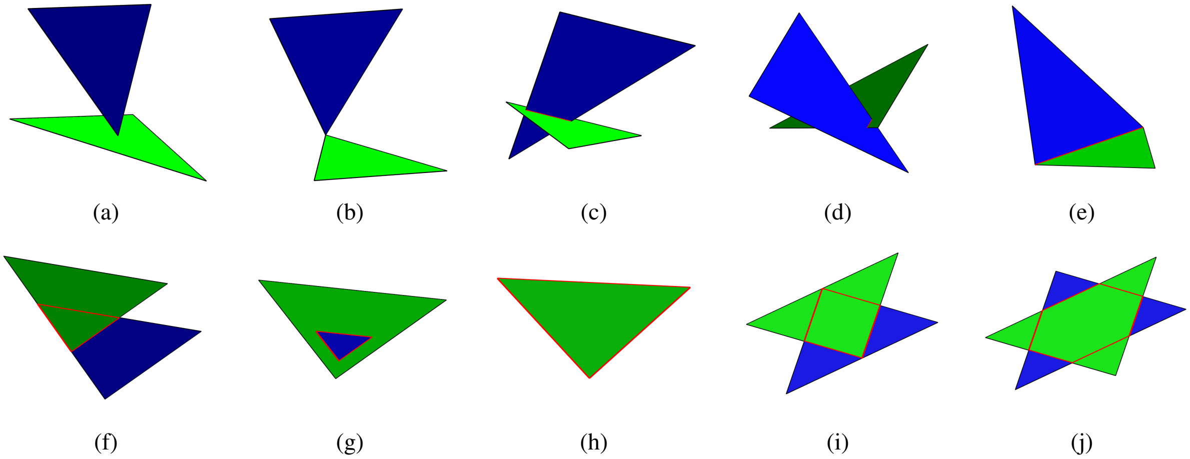


Figure 6: *Different intersection cases. Constraints (line segments) are shown in red. (a, b) Case 1. (c, d, e) Case 2. (f, g, h) Case 3. (i, j) Case 4. (b,e,h) represent singularities.*

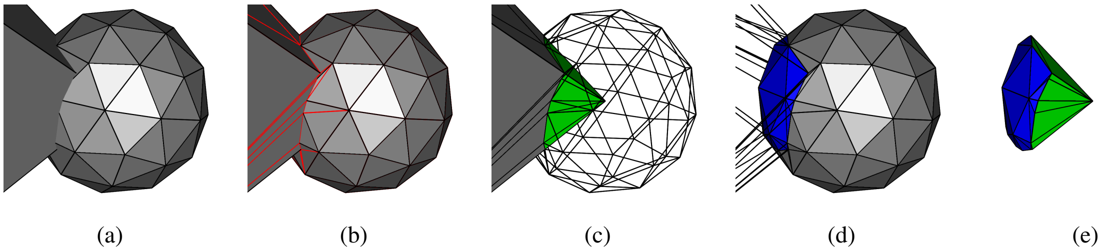


Figure 7: *Clipping algorithm. (a) Input: a Voronoi cell and a 3D object (here: a closed ball) bounded by a mesh. (b) Constrained Delaunay triangulations of the boundary of the cell and of the mesh. (c) In green: boundary of the cell inside the closed ball. (d) In blue: part of the mesh inside the cell. (e) Result: the clipped cell is bounded by the green and the blue triangulations.*

of the two parameters described in Section 4: initial number of sites and number of subdivisions. We compare the regularity of 2D and 3D CVTs generated with our hierarchical approach to CVTs computed with previous work (see Section 2.2). We also provide some details about computation time. In all figures, CVT cells Ω_i are colour-coded according to the cell regularity measure $G_m(\Omega_i)$ defined in Section 3: cells with a high dimensionless second moment are coloured in red, while cells with a low dimensionless second moment are coloured in blue.

5.1. Sensitivity to the initial number of sites

As stated in Section 4, the idea that drives our hierarchical approach is to first create a large regular area in a coarse tessellation, and then to preserve and possibly widen this area when subdividing. An example is shown in Figure 5. The average value of the cell regularity measure $G_2(\Omega_i)$ over all cells Ω_i expresses that the tessellation becomes more regular over subdivision, see Table 2. If n and s are large enough, we expect most of the interior cells of a CVT to be regular, see Figure 1 for an example.

CVT	(a)	(b)	(c)	(d)	(e)
Number of sites	10	40	160	640	2560
$\overline{G}_2(\Omega_i) \times 10^{-2}$	8.458	8.246	8.154	8.099	8.066

Table 2: *Average regularity measure for CVTs depicted in Figure 5* (1033 *sites, random sampling initialisation).*

On the same 2D square example, we create two other CVTs with 1033 and 2560 sites, respectively, using also random sampling initialisation and the same number of subdivisions ($s = 4$). We obtained an average cell regularity measure of 8.086×10^{-2} and 8.066×10^{-2}, respectively. This shows that the greater the number n of sites, the smaller the average cell regularity measure.

5.2. Sensitivity to the number of subdivisions

A CVT with a few big cells is likely to contain less different regular areas than a CVT of the same object with more, thus smaller, cells. Since our subdivision scheme preserves regular areas, a CVT generated with a large number s of subdivisions is more regular than a CVT with the same number n of sites but generated with a small s, as shown in Table 3. As a consequence, we suggest to set s as large as possible.

s	0	1	2	3	4
$\overline{G}_2(\Omega_i) \times 10^{-2}$	8.079	8.063	8.060	8.056	8.054

Table 3: *Average regularity measure for CVTs of a square with* 10000 *sites generated using our hierarchical approach (random sampling initialisation) with different number of subdivisions.*

In the case that an optimal lattice sampling is used as initialisation, it is preferable to start from a small number of subdivisions, since we only have one large regular area whatever the lattice size. Actually, $s = 0$ correspond to the optimal lattice sampling, as shown for instance in Figures 8 (f) and 9 (i). However, as stated before, the larger n, the more difficult it is to build such a lattice with a prescribed number of sites.

5.3. Comparison to previous work

We test our approach against previously mentioned methods in a simple 2D square. To check which method gives the most regular CVT, we first compute an hexagonal lattice with approximately 1000 sites. As stated above, it is difficult to accurately set the number n of sites. We were able to set $n = 1033$. We then compute CVTs with 1033 sites using the following methods:

- random sampling and L-BFGS update;
- Hammersley sampling [QSL*12] and L-BFGS update;
- global Monte-Carlo optimisation [LSPW12];
- our hierarchical approach with random sampling initialisation step (4 subdivisions, which is the maximum possible);
- our hierarchical approach with a lattice sampling initialisation step (1 subdivision).

For the global Monte-Carlo optimisation, we have used the parameter values advised in [LSPW12]. In particular, 200 updates have been performed.

Qualitative results are shown on Figure 8. The average values of the cell regularity measure $G_2(\Omega_i)$ over all cells Ω_i, as well as the CVT energy function values, are given in Table 4. Remember that, as explained in Section 3, an optimal cell has a dimensionless second moment value of $G_2 = \frac{5}{36\sqrt{3}} = 0.08018\ldots$

The hexagonal lattice is not optimal because of its boundary cells which are not hexagonal. Among other methods, the stochastic approach of [LSPW12] and our hierarchical framework give similar results. The main difference in practice between these two methods is the computation time: the global Monte-Carlo minimisation is about 100 times slower than our approach (207.08 seconds instead of 2.16).

We then test a geometrically more complex object: a five-branches star. We also increase the number of sites. As in the previous case, we first start by computing an hexagonal lattice with approximately 10000 sites. The lattice contains exactly $n = 10025$ sites. We discard the stochastic approach of [LSPW12] since it is too slow in this case. For the Hammersley sampling, we construct a bounding box of the object, and try different numbers of sites until we exactly obtain 10025 sites inside the object. It must be mentioned that several attempts were necessary since the number of sites inside the object does not necessarily increase when more sites are generated in the bounding box. For the hierarchical approach, we use 5 subdivisions after the random sampling initialisation (the maximum possible) and only one after lattice sampling initialisation.

Results are shown on Figure 9 and in Table 5. These results are in accordance with the results for the square. Our hierarchical approach performs better than the previous initialisation methods for both criteria. Moreover, lattice sampling initialisation gives better results than random sampling initialisation by almost reaching the regularity of a clipped hexagonal lattice.

In Figure 10 and Table 6 we compare CVTs computed on a simple closed 3D ball using random sampling, Hammersley sampling, our hierarchical approach with random sampling initialisation and our hierarchical approach with lattice sampling initialisation. It was not possible, in this case, to construct a BCC lattice with a prescribed and sufficiently large number of sites. Two different numbers n of sites were tested. We used the maximum number of subdivisions for the hierarchical approach with random sampling initialisation: two in the $n = 1000$ sites case and three in the $n = 5000$ sites case.

Remember that in this 3D case, the optimal cell regularity measure value is $G_3 = \frac{19}{192\sqrt[3]{2}} = 0.0785433\ldots$ This example shows that our hierarchical approach performs better than other initialisation methods, in case the number n of sites is high enough. In case not, the number of boundary cells is too high with respect to the number of interior cells for the Gersho's conjecture to apply in practice.

Other 3D CVTs with 50k, 80k and 100k sites, computed with a standard random sampling initialisation + L-BFGS update, our hierarchical approach using a random sampling initialisation and our hierarchical approach with a lattice sampling initialisation, are shown in Figure 11. It was not possible to create an Hammersley initialisation and a BCC lattice with the correct number of sites inside the objects in these cases.

5.4. Convergence speed

The main parameter in most CVT computation methods is the number of iterations of the algorithm. A local minimum of the CVT energy function is asymptotically reached, but

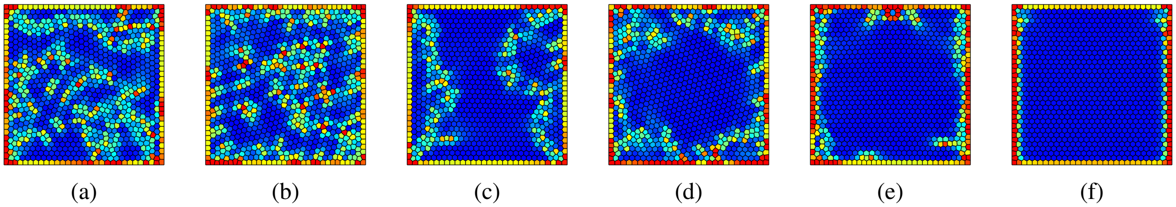


Figure 8: *CVTs with* 1033 *sites. (a) Random sampling + L-BFGS update. (b) Hammersley sampling [QSL*12] + L-BFGS update. (c) Global Monte-Carlo [LSPW12]. (d) Our approach, random sampling initialisation. (e) Our approach, lattice sampling initialisation. (f) Hexagonal lattice.*

CVT	(a)	(b)	(c)	(d)	(e)	(f)
$\overline{G}_2(\Omega_i)$	0.08101	0.08107	0.08075	0.08086	0.08077	0.08066
Energy function	25.140	25.160	25.041	25.082	25.045	25.002

Table 4: *Average regularity measure and CVT energy function value for CVTs depicted in Figure 8.*

little is known about how many iterations are necessary before convergence. We decided to investigate this for the previously described methods (except the global Monte-Carlo minimisation of [LSPW12]). The evolution of both the average of cell regularity measures $\overline{G}_2 = \frac{1}{n}\sum_{i=1}^{n} G_2(\Omega_i)$ and the CVT energy function value with respect to the number of iterations of the CVT L-BFGS update are displayed in Figure 12 for the star shape. For our hierarchical approach, this means the number of iterations of the last update (updates for coarser CVTs were done until the usual stopping criterion $\frac{\|g\|}{\|X\|} < 10^{-10}$ is reached, with g the gradient and X the vector of site coordinates, see [QSL*12]).

As shown in Figure 12, all methods behave the same for both measures. This was expected in our case (see Section 3.3). The hexagonal lattice converges the fastest and to the smallest value. Then our hierarchical approach, combined with a lattice sampling initialisation, gives the best results. It is interesting to notice that, because of the non hexagonal boundary cells, no approach reaches the theoretical optimal values ($G_2 = 0.08018\ldots$ and $E_2 = 0.221\ldots$, computed using Eq. (4)).

5.5. Computation time

Our algorithm is implemented in C++ and we use the CGAL library [cga] for 2D constrained and 3D Delaunay triangulations, and the libLBFGS library [ON10] for L-BFGS computation. All computations were performed on an Intel Xeon E5-2643 with 3.30 GHz CPU.

Computation times for our clipping method are shown in Table 7. In particular, we have tested this method on complex and badly triangulated objects and scenes to show its efficiency and robustness, see Figure 13.

Computation times for our hierarchical approach with random sampling initialisation (4 subdivisions) and for a standard method combining a random sampling initialisation and L-BFGS updates are shown in Table 8. Both methods are computationally equivalent in 2D, but ours is faster in 3D. This can be explained by the fact that in our approach the first CVT is computed with a small number of sites, which is very fast, while the next ones quickly converge since most of the sites do not move much.

Object	Fig.	#V (k)	#T (k)	Sites (k)	Time (s)
Homer	1	10	20	35	28.66
Ball	10	0.5	1	1	0.55
Bunny	11	10	20	50	30.06
Kitten	11	10	20	80	54.57
Armadillo	11	173	346	100	103.06
Ballgame	13	12.4	24.8	10	17.92
Dancer	13	15.1	30.2	5	10.52
Dragon	13	100	200	0.1	1.89
Dragon	13	100	200	1	6.14
Dragon	13	100	200	10	21.26
Dragon	13	100	200	100	99.09
CAD model	13	182	364	100	269.67

Table 7: *Computation times for clipped Voronoi diagrams.*

Other initialisation methods (Hammersley sampling and lattice sampling) are usually more time-consuming since finding the right density for a given number n of sites inside the object is difficult in practice. As stated before, the computation time for the stochastic approach of [LSPW12] depends on the number K of perturbations allowed. For a standard value $K = 200$, we found it to be very time consuming ($207.08s$ in the case of the 2D square with $n = 1000$ sites).

6. Conclusions and Future Work

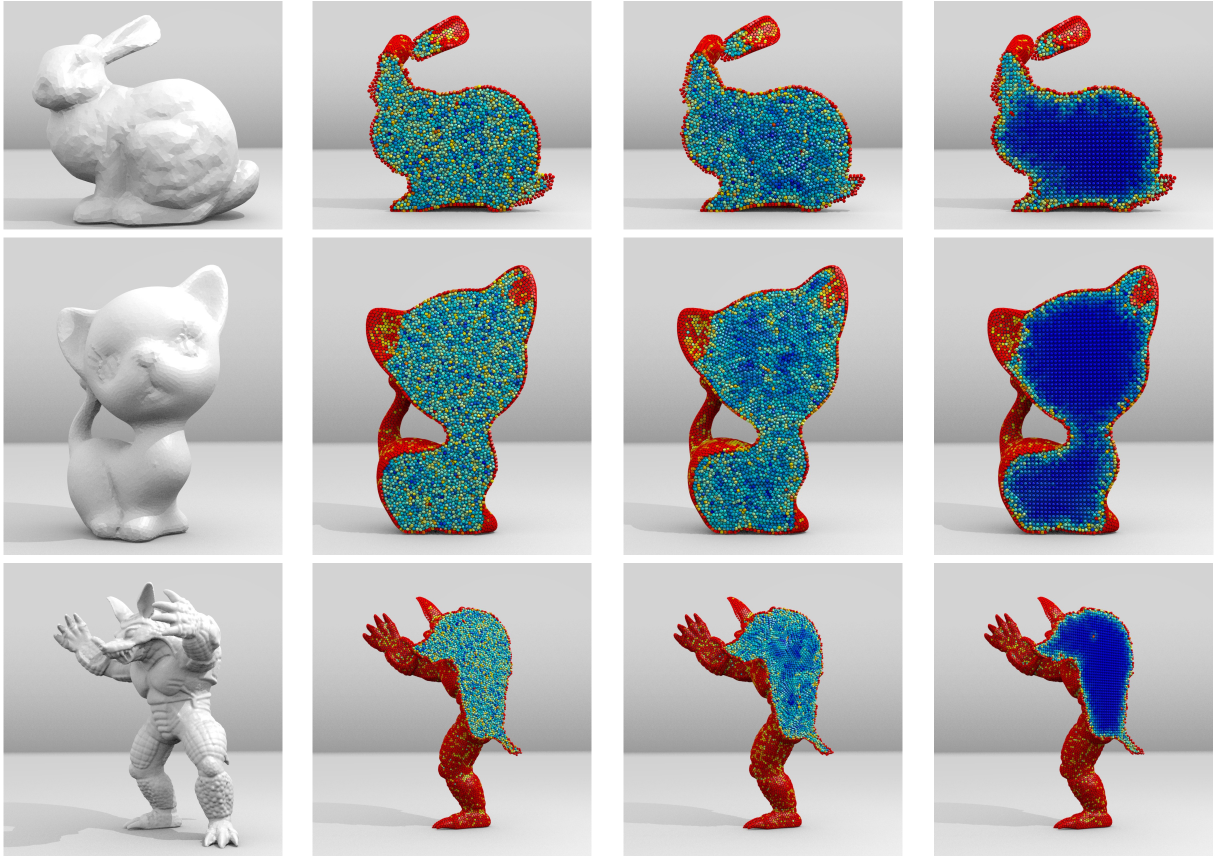
We have introduced different contributions to CVT in two and three dimensional spaces. A regularity criterion was de-

CVT	(a)	(c)	(e)	(g)	(i)
$\overline{G}_2(\Omega_i) \times 10^{-2}$	8.089	8.074	8.068	8.054	8.049
Energy function $\times 10^{-1}$	2.234	2.229	2.227	2.222	2.220

Table 5: *Average regularity measure and CVT energy function value for CVTs depicted in Figure 9.*

CVT	(a)	(c)	(e)	(g)	(b)	(d)	(f)	(h)
$\overline{G}_3(\Omega_i) \times 10^{-2}$	8.025	8.025	8.022	8.022	7.980	7.976	7.975	7.965
Energy function $\times 10^{-3}$	4.324	4.325	4.324	4.325	1.471	1.470	1.470	1.468

Table 6: *Average regularity measure and CVT energy function value for CVTs depicted in Figure 10.*



1

Figure 11: *More examples of comparaisons between a standard approach and our hierarchical one. From left to right: Input object, Random sampling + L-BFGS update, Our approach with random sampling initialisation, Our approach with lattice initialisation.*

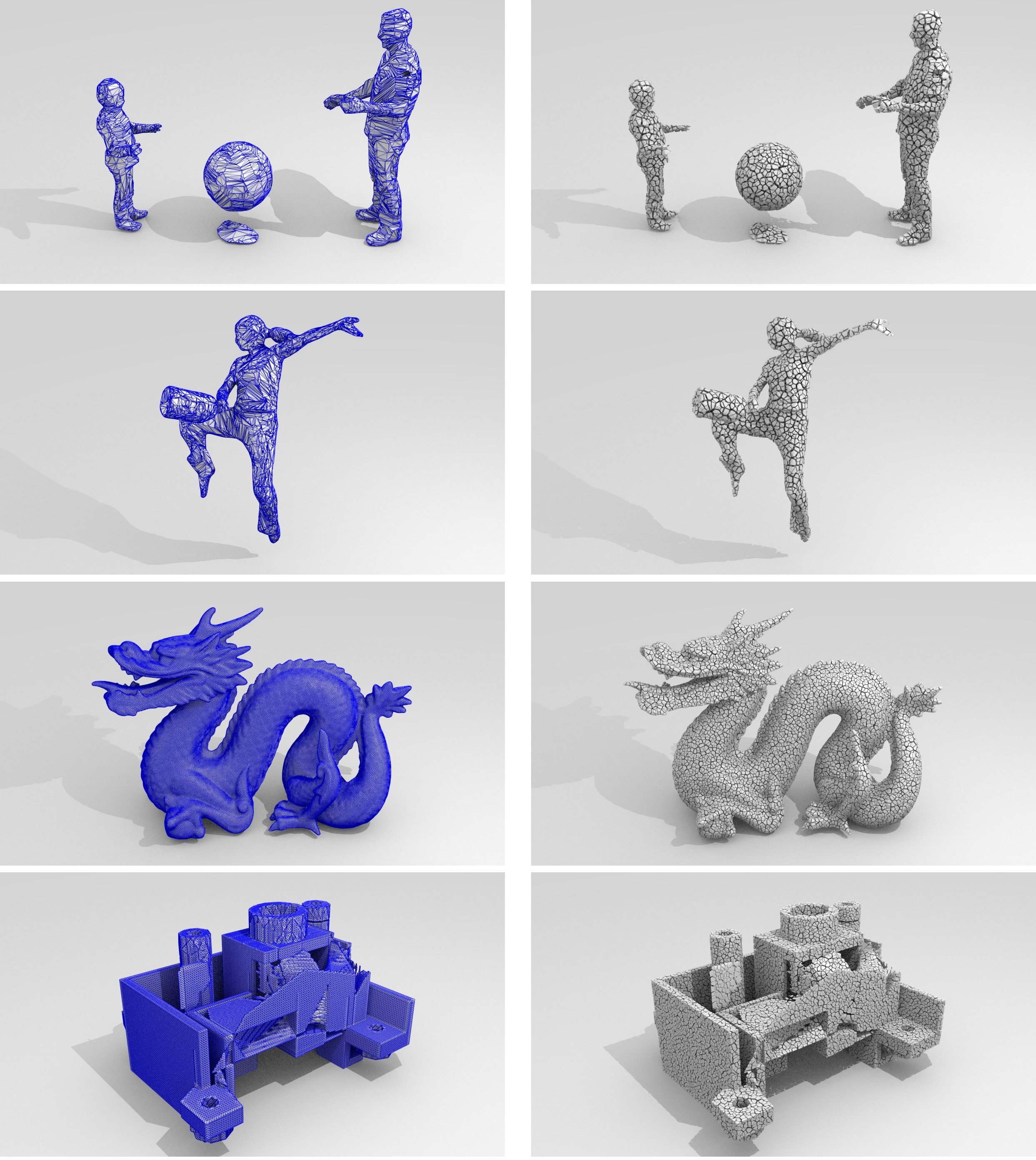
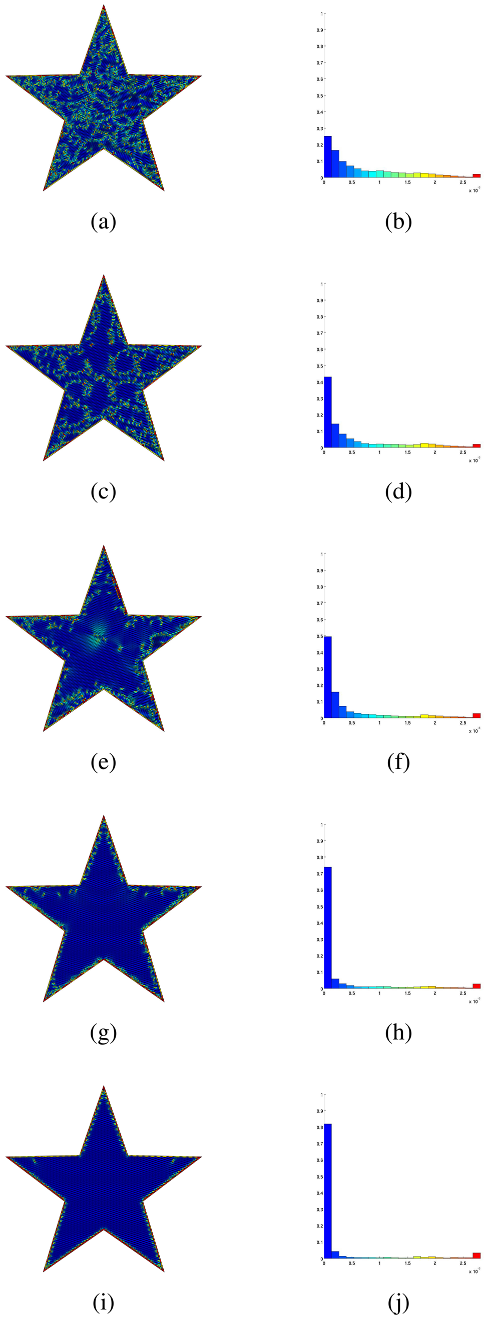


Figure 13: *More examples of clipped (non Centroidal) Voronoi diagrams. Left: input triangulations. Right: clipped Voronoi diagrams.*



(a) (b)

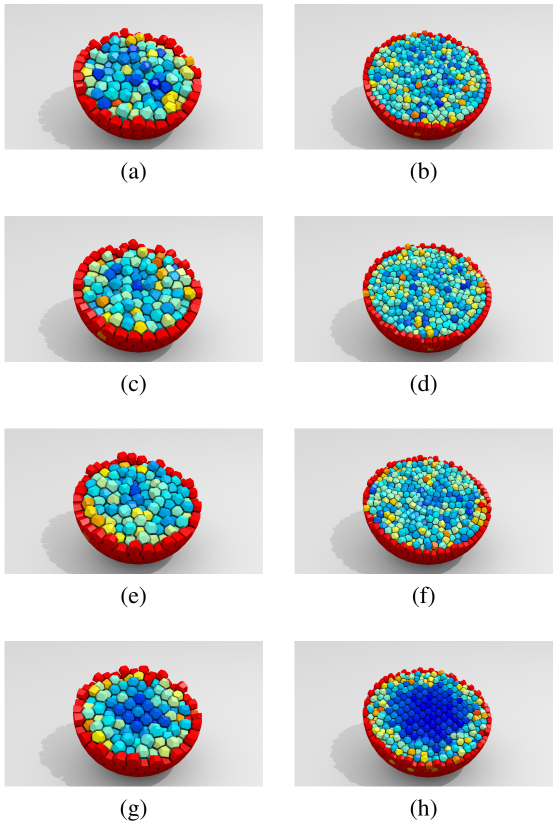
(c) (d)

(e) (f)

(g) (h)

(i) (j)

Figure 9: *(a,c,e,g,i) CVTs with* 10025 *sites. (b,d,f,h,j) Corresponding regularity histograms: each bin indicates how many cells share a regularity measure comprised between its boundary values. (a,b) Random sampling + L-BFGS update. (c,d) Hammersley sampling [QSL*12] + L-BFGS update. (e,f) Our approach, random sampling initialisation. (g,h) Our approach, lattice sampling initialisation. (i,j) Hexagonal lattice.*



(a) (b)

(c) (d)

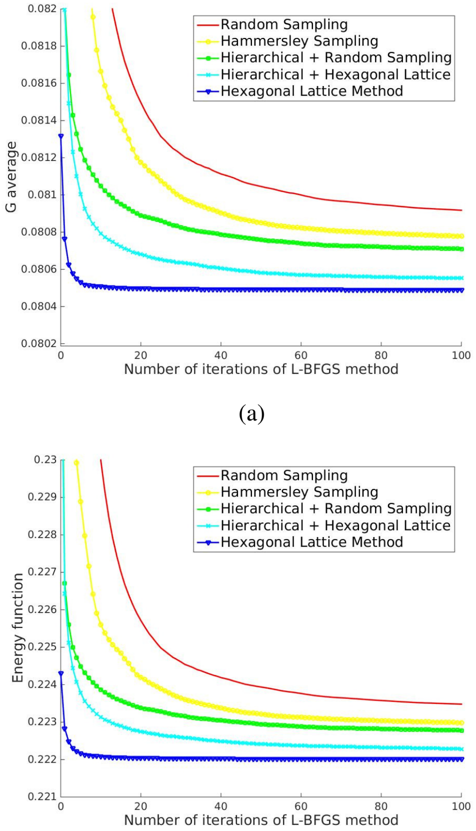
(e) (f)

(g) (h)

Figure 10: *(a,c,e,g) CVTs with* 1000 *sites in a sphere. (b,d,f,h) CVTs with* 5000 *sites. (a,b) Random sampling + L-BFGS update. (c,d) Hammersley sampling [QSL*12] + L-BFGS update. (e,f) Our approach, random sampling initialisation. (g,h) Our approach, lattice sampling initialisation.*

Object	Fig.	Sites	Method	Time (s)
Square	8	1000	Standard	2.84
			Hierarchical	2.16
Square	8	5000	Standard	10.85
			Hierarchical	9.50
Star	9	2000	Standard	3.56
			Hierarchical	4.91
Star	9	10000	Standard	26.94
			Hierarchical	26.97
Ball	10	1000	Standard	521.86
			Hierarchical	205.82
Ball	10	5000	Standard	1484.69
			Hierarchical	665.01
Homer	1	50000	Standard	15720
			Hierarchical	7860

Table 8: *Computation times for CVTs of objects depicted in Figures 8, 9, 10 and 1.*



(a)

(b)

Figure 12: *Average cell regularity (a) and CVT energy function value (b) with respect to the number of iterations of the CVT update, for the star shape displayed in Figure 9.*

fined to evaluate the quality of CVTs. We also proposed a hierarchical approach for generating CVTs with increased regularities with respect to existing methods as well as a new solution to clip Voronoi tessellations in 3D. Our approach can be used for CVTs in higher dimensional spaces, combined with an adapted Voronoi clipping algorithm such as [LÍ4], although Gersho's conjecture has not been proven in this case. In future work, we may also consider extensions of the approach to generalised CVTs such as for instance weighted diagrams, power diagrams or L_p CVTs [LL10].

Acknowledgements

This work is partly supported by the ANR (project ANR-10-BLAN-0206 MORPHO). We would like to thank Lin Lu for the code of [LSPW12], and Bruno Lévy and the reviewers for their remarks. The CAD model (Figure 13) is courtesy of Jean-Claude Léon.

References

[ACSYD05] ALLIEZ P., COHEN-STEINER D., YVINEC M., DESBRUN M.: Variational tetrahedral meshing. *ACM Transactions on Graphics 24* (2005), 617–625. 3

[AFB15] ALLAIN B., FRANCO J.-S., BOYER E.: An Efficient Volumetric Framework for Shape Tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)* (2015). 2

[Aur91] AURENHAMMER F.: Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR) 23* (1991), 345–405. 2

[BS83] BARNES E. S., SLOANE N. J. A.: The optimal lattice quantizer in three dimensions. *SIAM Journal on Algebraic Discrete Methods 4*, 1 (1983). 3

[BSD09] BALZER M., SCHLÖMER T., DEUSSEN O.: Capacity-constrained point distributions: A variant of lloyd's method. *ACM Transactions on Graphics 28*, 3 (2009), 86:1–86:8. doi:10.1145/1531326.1531392. 2

[cga] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org. 10

[CS82] CONWAY J. H., SLOANE N. J. A.: Voronoi regions of lattices, second moments of polytopes, and quantization. *IEEE Transactions on Information Theory 28* (1982), 211–226. 2, 4

[DE06] DU Q., EMELIANENKO M.: Acceleration schemes for computing centroidal voronoi tessellations. *Numerical Linear Algebra with Applications 13* (2006), 173–192. 3

[DFG99] DU Q., FABER V., GUNZBURGER M.: Centroidal voronoi tessellations: Applications and algorithms. *SIAM review 41* (1999), 637–676. 2, 3

[Eri03] ERICKSON J.: Nice point sets can have nasty delaunay triangulations. *Discrete & Computational Geometry 30*, 1 (2003), 109–132. doi:10.1007/s00454-003-2927-4. 6

[ES94] EDELSBRUNNER H., SHAH N. R.: Triangulating topological spaces. In *Proceedings of the tenth annual symposium on Computational geometry* (1994), pp. 285–292. 6

[For92] FORTUNE S.: Voronoi diagrams and delaunay triangulations. *Computing in Euclidean geometry 1* (1992), 193–233. 2

[Ger79] GERSHO A.: Asymptotically optimal block quantization. *IEEE Transactions on Information Theory 25* (1979), 373–380. 3

[GG91] GERSHO A., GRAY R. M.: *Vector quantization and signal compression*. Kluwer Academic Publishers, 1991. 4

[JRG11] JU L., RINGLER T., GUNZBURGER M.: Voronoi tessellations and their application to climate and global modeling. In *Numerical Techniques for Global Atmospheric Models*, Lauritzen P., Jablonowski C., Taylor M., Nair R., (Eds.). Springer, 2011, pp. 313–342. 2

[LÍ4] LÉVY B.: Restricted voronoi diagrams for (re)-meshing surfaces and volumes. In *8th International Conference on Curves and Surfaces* (2014). 6, 14

[LL10] LÉVY B., LIU Y.: Lp centroidal voronoi tessellation and its applications. *ACM Transactions on Graphics 29*, 119 (2010). 14

[Llo82] LLOYD S. P.: Least squares quantization in pcm. *IEEE Transactions on Information Theory 28* (1982), 129–137. 3

[LSPW12] LU L., SUN F., PAN H., WANG W.: Global optimization of centroidal voronoi tessellation with monte carlo approach. *IEEE Transactions on Visualization and Computer Graphics 18* (2012), 1880–1890. 2, 3, 4, 9, 10, 14

[LWL*09] Liu Y., Wang W., Lévy B., Sun F., Yan D.-M., Liu L., Yang C.: On centroidal voronoi tessellation - energy smoothness and fast computation. *ACM Transactions on Graphics 28*, 101 (2009). 2, 3, 6

[MS06] Moriguchi M., Sugihara K.: A new initialization method for constructing centroidal voronoi tessellations on surface meshes. In *3rd International Symposium on Voronoi Diagrams in Science and Engineering, 2006* (2006), pp. 159–165. 3

[New82] Newman D. J.: The hexagon theorem. *IEEE Transactions on Information Theory 28* (1982), 137–139. 3

[OBSC00] Okabe A., Boots B., Sugihara K., Chi S. N.: *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley, 2000. 2

[ON10] Okazaki N., Nocedal J.: libLBFGS: a library of Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS), 2010. http://www.chokkan.org/software/liblbfgs/. 10

[QSL*12] Quinn J., Sun F., Langbein F. C., Lai Y.-K., Wang W., Martin R. R.: Improved initialisation for centroidal voronoi tessellation and optimal delaunay triangulation. *Computer-Aided Design 44* (2012), 1062–1071. 2, 3, 7, 9, 10, 13

[SH74] Sutherland I. E., Hodgman G. W.: Reentrant polygon clipping. *Communications of the ACM 17* (1974), 32–42. 6

[She98] Shewchuk J. R.: A condition guaranteeing the existence of higher-dimensional constrained delaunay triangulations. In *SCG '98 Proceedings of the fourteenth annual symposium on Computational geometry* (1998), pp. 76–85. 6

[She08] Shewchuk J. R.: General-dimensional constrained delaunay and constrained regular triangulations, i: Combinatorial properties. *Discrete & Computational Geometry 39* (2008), 580–637. 6

[TAD07] Tournois J., Alliez P., Devillers O.: Interleaving delaunay refinement and optimization for 2d triangle mesh generation. In *Proceedings of the 16th International Meshing Roundtable* (2007), pp. 83–101. doi:10.1007/978-3-540-75103-8_5. 4

[TWAD09] Tournois J., Wormser C., Alliez P., Desbrun M.: Interleaving delaunay refinement and optimization for practical isotropic tetrahedron mesh generation. *ACM Transactions on Graphics 28*, 75 (2009). 4

[YLL*09] Yan D.-M., Lévy B., Liu Y., Sun F., Wang W.: Isotropic remeshing with fast and exact computation of restricted voronoi diagram. *Computer Graphics Forum 28* (2009), 1445–1454. 6

[YWLL13] Yan D.-M., Wang W., Lévy B., Liu Y.: Efficient computation of 3d clipped voronoi diagram for mesh generation. *Computer-Aided Design 45* (2013), 843–852. 2, 6

[ZBH11] Zaharescu A., Boyer E., Horaud R.: Topology-adaptive mesh deformation for surface evolution, morphing, and multi-view reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33* (2011), 823–837. 6