Improved Initialisation for Centroidal Voronoi Tessellation and Optimal Delaunay Triangulation

Jonathan Quinn[a], Feng Sun[b], Frank C Langbein[a,*], Yu-Kun Lai[a], Wenping Wang[b], Ralph R Martin[a]

[a]*Cardiff University, School of Computer Science and Informatics, 5 The Parade, Cardiff, CF24 3AA, UK*
[b]*The University of Hong Kong, Pokfulam Road, Hong Kong*

Abstract

Centroidal Voronoi tessellations and optimal Delaunay triangulations can be approximated efficiently by non-linear optimisation algorithms. This paper demonstrates that the point distribution used to initialise the optimisation algorithms is important. Compared to conventional random initialisation, certain low-discrepancy point distributions help convergence towards more spatially regular results and require fewer iterations for planar and volumetric tessellations.

Keywords: Centroidal Voronoi Tessellation, Optimal Delaunay Triangulation, Low Discrepancy, Hammersley Sequence

1. Introduction

Centroidal Voronoi tessellations (CVTs) [1] and optimal Delaunay triangulations (ODTs) [2] are highly regular tessellations of an Euclidean region. Each cell within a tessellation contains a point sample, or site, to which the cell belongs. The properties of CVTs and ODTs mean that their sites are spatially very evenly distributed: as sites in a CVT or ODT tend to be close to a regular hexagonal lattice, triangulating them results in a large proportion of equilateral triangles. Such regular arrangements are highly desirable for many applications, such as solving partial differential equations using finite element analysis and the construction of response surfaces. Hence, CVT and ODT generation is a problem studied widely in geometric modelling and numerical analysis.

Given a set of sites in a Euclidean region, the Voronoi cell of a site is the subset of the region closest to the site. The set of all Voronoi cells is referred to as the Voronoi tessellation of the region. A CVT is a special type of Voronoi tessellation in which the site of each Voronoi cell is positioned at the centroid of its cell [1]. This results in a very regular tessellation: in a globally optimal 2D CVT, each Voronoi cell far from the boundary of the region converges to a regular hexagon as the number of sites goes to infinity [3]. An ODT is the triangulation of a Euclidean region that minimises the interpolation error of a given function [2]. Typically the function of a standard elliptic paraboloid is used. In this case, when the sites are fixed, the optimal connectivity is given by the Delaunay triangulation of these sites. Ignoring boundary effects, the interior triangles tend to be regular when the number of sites is sufficiently large. Given a fixed number of samples, the globally optimal distributed sites for CVTs and ODTs have a deterministic layout. In practice, a good approximation can be obtained with an optimisation method which iteratively updates some initial distribution of sites. The ODTs and CVTs resulting from this have slightly different tessellation properties: when triangulated, CVT approximation produces more nearly regular triangles in 2D meshes, while ODT approximation results in fewer slivers (degenerate tetrahedrons [4]) in 3D meshes [5].

Point distributions produced by CVT and ODT methods are uniformly dense and isotropic and, locally, neighbouring samples are very evenly spaced [6]. For geometric modelling and graphics applications, such sampling is very useful as indicated by the blue noise criterion: there are minimal spikes of energy in the mid to high radial frequencies, meaning that no additional structure is introduced by the sampling process [7]. Simplicies generated from the sampling are also close to equilateral, minimising numerical problems. Therefore, CVT and ODT approximations have been applied to various problems, including remeshing [5, 8], dithering and finite element mesh generation.

CVT generation relies on probabilistic iterative solutions such as Lloyd's method [9]. Liu et al. [8] proved that the piecewise CVT energy function is C^2 continuous for convex, compact, regions. Hence, it can locally be optimised by non-linear numerical optimisation methods. Consequently they use limited-memory BFGS (L-BFGS) to improve the efficiency of CVT computation. They also describe methods to compute a density-controlled CVT (where the metric is locally scaled with respect to a density function) using L-BFGS and preconditioned L-BFGS (P-L-BFGS), yielding fast convergence even for large-scale problems. Efficient ODT computation uses Chen's method [2], which is similar to Lloyd's iterative CVT method and has been further improved by Alliez et al. [5].

There are few results on improving the efficiency of the optimisation methods or the regularity of the output sample distribution by carefully choosing the initial site distribution.

*Phone: +44-29-208-70110, Fax: +44-29-208-74598
Email addresses: j.a.quinn@cs.cf.ac.uk (Jonathan Quinn), fsun@cs.hku.hk (Feng Sun), LangbeinFC@cf.ac.uk (Frank C Langbein), yukun.lai@cs.cf.ac.uk (Yu-Kun Lai), wenping@cs.hku.hk (Wenping Wang), r.r.martin@cs.cf.ac.uk (Ralph R Martin)

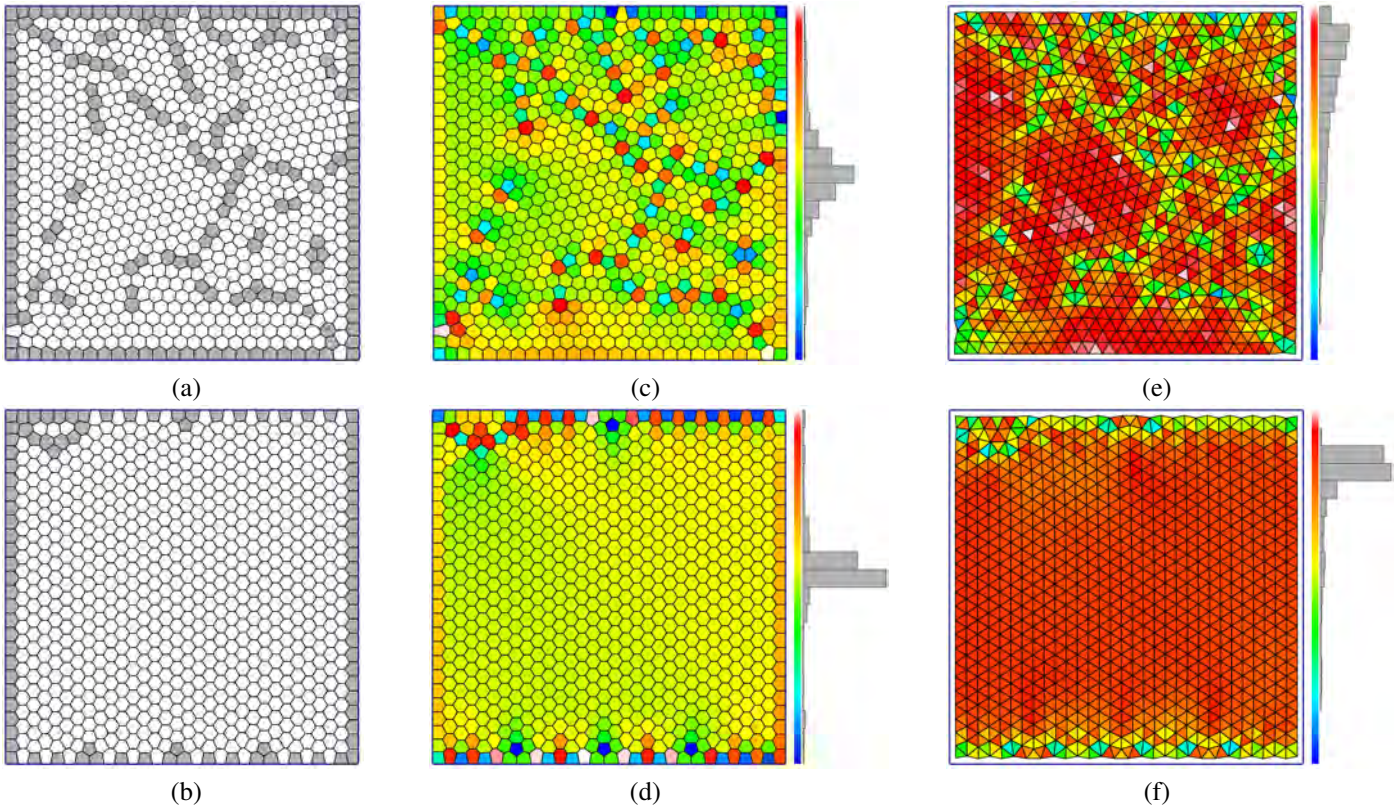

Figure 1: Two CVTs with 800 seeds in a square. Top row: random initialisation converging to an energy of 2.43704×10^{-3}. Bottom row: Hammersley initialisation converging to an energy of 2.43135×10^{-3}. (a) and (b) show non-hexagonal cells in the CVT highlighted in grey. While the difference of the energy is minor, there is a significant improvement in the regularity of the Voronoi cells with the Hammersley initialisation when compared with the random initialisation. (c) and (d) show the energy of the sites increasing with colour from blue to red to white. The histogram of the energy in each cell is illustrated on the right side and shows that the energy with the Hammersley initialisation is more concentrated, mostly in the middle two bars. (e) and (f) show the dual triangulation of the CVT with colour coding of the smallest angle in each triangle: blue indicates 40° and red indicates 60°. The Hammersley initialisation improves the statistics of smallest angles significantly.

Moriguchi [10] describes a method to improve the initialisation for CVTs by performing a greedy edge-collapsing decimation on the input mesh and uses the vertices from the decimated mesh as the set of input sites. It seems that this method reduces the number of iterations required to reach the same energy value. However, as results are only shown for 20 iterations of Lloyd's method on a single example mesh, it is difficult to evaluate how this initialisation behaves in general. Performing mesh decimation to generate initialisation sites also requires considerable computational effort, which may nullify any speed improvement introduced during the optimisation process. Moreover, it is not directly applicable to general boundary representations or parametric surfaces. Finally, if the input for this method is not a regular mesh, even the greedy edge-collapsing method does not guarantee a regular distribution of sites. In fact, if the number of sites required is similar to the density of the input mesh – a situation common in remeshing – the resulting initialisation of sites may be extremely poor, making the method worse than random initialisation.

Intuitively, if the initialisation sites are close to the sites after convergence, fewer iterations of the method will be required. Thus, as the sites in a CVT and ODT are very regularly placed, we note that using a more regularly spaced set of initialisation sites has two advantages over a random distribution. Firstly, it reduces the number of iterations required to achieve convergence. Secondly, it allows the optimisation process to converge to a more spatially regular set. In particular, a random sampling tends to result in several regions of quite-regular hexagon tessellation, separated by boundaries. Across these boundaries, there exist clear mismatches in position and orientation between the hexagonal tilings. In contrast, the initialisation approach introduced here tends to produce far fewer boundaries, and often provides a result comprising a single region. For example, see Fig. 1, discussed in detail in Section 5, for the CVT of a square generated using random and low-discrepancy initialisation.

Low-discrepancy, or quasi-Monte Carlo, sequences, whilst not as spatially regular as CVT and ODT site distributions, are point sets that have the lowest possible order of magnitude of discrepancy for an Euclidean region [11]. This means that whilst they may have high levels of structure in the sampling, as seen in radially averaged power spectrum density measures [12], they cover a region with a highly uniform density, resulting in very few 'holes' in the distribution. In addition, they are very fast and simple to compute. Thus, in this paper we investigate how they may be used as initialisation sites for CVT and ODT methods in 2D and 3D Euclidean space, for

both uniform and user-defined site density functions. We focus in particular on CVT in 2D and then consider more general cases.

In the remainder of this paper, we summarise CVT and ODT generation methods (Section 2) and low-discrepancy sequences (Section 3). We then show how to redistribute the sites in these sequences to match a specified density (Section 4), and evaluate and compare CVT and ODT generation with random and low-discrepancy initialisation (Section 5). Finally, we draw conclusions in Section 6.

2. Computation of CVT and ODT

In this section, we introduce CVT and ODT in more detail and describe the algorithms used for their computation.

2.1. Computation of CVT

The inputs for CVT are a set of sites $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) \in \mathbb{R}^d$, to be distributed with respect to a density function $\delta > 0$ in the compact region $\mathbf{\Omega} \subset \mathbb{R}^{\mathbf{d}}$. We define the CVT energy function CVT as [1]

$$F(\mathbf{X}) = \sum_{i=1}^{n} F_i(\mathbf{X}) = \sum_{i=1}^{n} \int_{\mathbf{\Omega_i}} \delta(\mathbf{x}) \|\mathbf{x} - \mathbf{x}_i\|^2 d\sigma \qquad (1)$$

where $\mathbf{\Omega_i}$ is the intersection of $\mathbf{\Omega}$ and V_i, i.e. the Voronoi cell of site $\mathbf{x}_i$. Minimising $F(\mathbf{X})$ ensures that each subregion $\mathbf{\Omega_i}$, and thus each site $\mathbf{x}_i$, represents approximately the same sub-volume of $\mathbf{\Omega}$ in the uniform density case. The properties and computation of CVT have been well studied [1, 13] and several algorithms, including Lloyd's method [9], the Lloyd-Newton method [14] and the Quasi-Newton method [8] have been proposed for computing a CVT for a given region. We briefly review these methods for the reader's benefit.

2.1.1. Lloyd's Method

Lloyd's method [9], the prevailing method before Liu et al. [8], introduced a quasi-Newton method for CVT computation. As an iterative method, in each iteration, Lloyd's method moves each seed $\mathbf{x}_i$ of the Voronoi region Ω_i to the centroid $\mathbf{c}_i$ of Ω_i, followed by updating the Voronoi tessellation of the seeds.

2.1.2. Lloyd-Newton Method

Du and Emelianenko [14] proposed an algorithm for computing a CVT by solving the system of equations $\mathbf{x}_i = \mathbf{c}_i$, $i = 1, \ldots, n$, iteratively. As pointed out by Liu et al. [8], this is equivalent to minimising the function $F = \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{c}_i\|^2$ and the result is not always a CVT since the minimisation may get stuck at a non-zero local minimum of F, in which case the $\mathbf{x}_i$ do not coincide with the centroids $\mathbf{c}_i$, $i = 1, \ldots, n$.

2.1.3. Quasi-Newton Method

Liu et al. [8] prove that the CVT function $F(\mathbf{X})$ is almost everywhere C^2, except for some configurations seldom met in practice. Based on the C^2 property, they apply a quasi-Newton method, the P-L-BFGS method, to minimise the CVT function $F(\mathbf{X})$ directly. This method uses the Hessian when available, constructed from the gradients of previous iterations. This is the fastest method currently available.

2.2. Computation of ODT

Compared to CVT, ODT is a more recent concept, proposed by Chen and Xu [2]. A computational framework is proposed in the same paper, which is the only existing method for computing ODT.

2.2.1. Chen's Method

The framework of Chen's method is similar to Lloyd's method; it is also iterative. In each iteration, two steps are performed: (1) compute the Delaunay triangulation of the seeds; (2) move the seed $\mathbf{x}_i$ to

$$\frac{\Sigma_{\mathbf{x}_i \in \mathbf{S}_j} \mathrm{Vol}(\mathbf{S}_j) \mathbf{c}(\mathbf{S}_j)}{\Sigma_{\mathbf{x}_i \in \mathbf{S}_j} \mathrm{Vol}(\mathbf{S}_j)},$$

where $\mathbf{S}$ is a d-dimensional simplex for a d-dimensional problem and $\mathbf{c}(\mathbf{S})$ is the centre of the circumsphere of the simplex $\mathbf{S}$. In step (2) the algorithm moves the seed to a weighted combination of the centres of circumspheres of all simplices adjacent to the seed. To assure the monotonic decrease of the objective function, in each iteration, only one vertex is moved and then the Delaunay triangulation is immediately updated after the movement. This algorithm is extremely slow. Alliez et al. [5] enhance it by moving all vertices in one iteration.

2.3. Summary of Existing Methods

Lloyd's method and the quasi-Newton method both generate CVTs; the latter represents the state of the art and is hence used in this paper. For ODT we choose the enhanced algorithm of Alliez et al [5] for efficient computation. In this paper we investigate the effect of random and Hammersley initialisation for both of these methods based on timing and regularity of the results.

3. Low-discrepancy Sequences

In this section we introduce the notion of discrepancy, the low-discrepancy sequences used in this work, and how we apply them to CVT and ODT initialisation. A uniform distribution is one in which any position for a point is equally likely: their positions are chosen according to a uniform probability distribution; there is an equal likelihood for a single point being at any position. However, uniformity only refers to the probability of each point considered individually, and thus, the position of these points is independent of the position of any other points, i.e. their position is uncorrelated. In practice, this often leads to areas of the domain which are over-sampled, and areas which are under-sampled; samples can appear clustered with large inter-cluster holes. Low-discrepancy sequences have correlated positions: the probability of a point being at some position is dependent on its location in the sequence and, in turn, the positions of its neighbours. This means that the quality, in particular the spatial coverage, of the whole point set is

considered, rather than a single, independent point, resulting in a more even coverage of a region. The spatial correlation can be measured by discrepancy, which allows us to quantitatively assess the size of gaps in the coverage. Sequences with low discrepancy are desirable for this reason.

Discrepancy became an important concept since it was demonstrated that as the discrepancy of a sequence decreases, so does the approximation error of a (quasi) Monte Carlo evaluation of a multivariate integral [15]; more importantly, the number of samples needed to achieve a given accuracy increases more slowly for low discrepancy samples than for random samples, as the desired accuracy increases. This observation has lead to the use of low discrepancy sequences in many different fields, such as computer graphics [16], surface representation [17], area computation [18] and volume computation [19]. The discrepancy $D^*(P)$ of a point set P of N points with respect to a function $f : \Omega \to \mathbb{R}$ can be thought of as the difference between the point-sampled numerical approximation of the integral $\int_\Omega f(x)\,dx$ and its actual value,

$$\left| \frac{1}{N} \sum_{x \in P} f(x) - \int_\Omega f(x)\,dx \right| \leq D^*(P)\mathrm{var}(f), \tag{2}$$

where $\mathrm{var}(f)$ is the variance of the integrand [20]. Discrepancy is often thought of as the Monte Carlo approximation error as defined by Eq. 2, but can also be loosely considered as a sequences deviation from a uniform sampling of a domain [11]. By Eq. 2, lowering the discrepancy $D^*(P)$ reduces the integral approximation error. Whilst there are various techniques to measure the discrepancy of a sample set, we use the common star discrepancy $D^*_\delta(P)$ of a set P with respect to a density function δ,

$$D^*_\delta(P) = \sup_{\gamma \subseteq \Omega} \left| \frac{|P \cap \gamma|}{N} - \frac{\int_\gamma \delta(x)dx}{\int_\Omega \delta(x)dx} \right|,$$

where Ω is the domain sampled with P and $|\cdot|$ the number of points in a discrete set. The subsets γ are usually restricted to axis-aligned rectangles in Ω [11], but other classes of shape may be used [12].

The Niederreiter and Sobol low-discrepancy sequences are believed to be optimal [11] for sampling axis-aligned rectangular regions, and use a lattice structure to enforce point distribution uniformity. However, they are generally avoided for situations where sampling with non-axis-aligned shapes is required [21]. The Hammersley [22] and Halton [23] sequences use the van der Corput sequence [24] for construction, and whilst having the lowest possible order of magnitude of discrepancy [11], they do not scale well in higher dimensions [11]. However, in low dimensions, they are more geometrically regular than the lattice-based methods.

Experimentally, we have found that the Hammersley sequence produces the best results for CVT and ODT initialisation, when compared to the Halton, Niederreiter and Sobol sequences, all three of which performed in a very similar way for CVT and ODT initialisation. For this reason, we only compare pseudorandom sampling with the Hammersley sequence and omit the other results here. Moreover, true random sampling was investigated, seeded using atmospheric noise [25], but was found to lead to similar results than pseudorandom sampling and therefore details are also omitted.

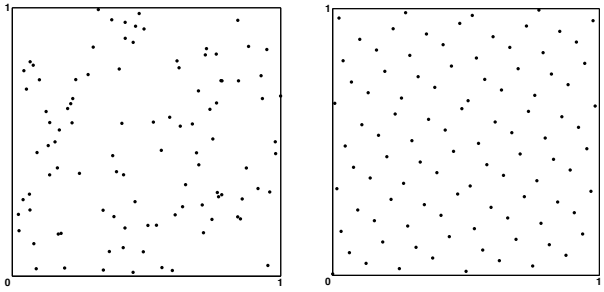

Figure 2: A pseudorandom distribution (left) and the Hammersley sequence (right), both shown with 100 point samples.

3.1. The Hammersley Sequence

The Hammersley low-discrepancy sequence (see Fig. 2) achieves the lowest possible order of magnitude of discrepancy [11] and also performed best for ODT and CVT initialisation in initial experiments. The van der Corput sequence, ψ_b, is a method to partition or sample the unit interval by maximising the distance between sample points, resulting in a uniform distribution on that interval. The premise is that a positive integer i can be expanded in base b by reflecting the b-ary representation of the digits, resulting in a fractional number in $[0, 1)$. The bases b_i are chosen pairwise co-prime. The Hammersley sequence is deterministic, but, as the first co-ordinate i/N depends on the size of the point set N, adding a single point alters the entire distribution. Thus, N cannot be increased incrementally, but must be defined prior to construction. For N samples, we compute a component of the co-ordinate using:

$$\psi_b(i) = \sum_{j=0}^{k-1} a_{ij} b^{-i-1}, \tag{3}$$

where $i = 0, \ldots, N-1$. a_{ij} represents the j'th bit of the b-ary representation of i, and b the base chosen for the sequence. In order to define the correct summation range, we compute the number of bits required for the b-ary representation, $k = \lceil \log_b N \rceil$, and thus $j = 0, \ldots, k-1$. This defines a d-dimensional Hammersley point:

$$p = (\tfrac{i}{N}, \psi_{b_1}(i), \ldots, \psi_{b_d}(i)). \tag{4}$$

4. Initialisation for CVT and ODT

In this section, we discuss the algorithms for generating random and Hammersley point samples, which we use as initialisation sites for CVT and ODT methods. In particular we discuss the initialisation for density-controlled CVT. Note that there is no method in the literature to generate an ODT with respect to a density function.

It is interesting to note that Romero et al. [6] briefly tried Hammersley points as an initialisation for CVT, but did not expect, and therefore notice, the improvement in performance and quality in their results.

Note that using a hexagonal grid to initialise the distribution instead of pseudorandom or Hammersley points would also be an option. However, generating such a distribution which places an exact number of sites in a given shape is quite difficult. In addition, due to the local optimality of the initial packing within a hexagonal grid, there is little possibility for seeds to change position, which may be a hindrance when sampling shapes with complex boundaries (especially as the gradient used in the optimisation may become too small, resulting in immediate convergence). Moreover, using a regular grid ceases to be useful when a non-uniform density function is desired, as there is no simple way to generalise it to such a case. Also, generating a regular grid is not computationally cheaper than generating Hammersley points.

The k-means algorithm [9] is the discrete analog of the Lloyd method for clustering n points into k clusters; the clusters partition these n points into k subsets. The k-means algorithm minimises the sum of the distance from each point to the centroid of the cluster it belongs to. Previously, several attempts at careful initialisation [26, 27] have been made to improve the k-means algorithm. However, these methods apply only to a discrete set of n points, but not a continuous domain and they are oblivious to the geometric properties of CVT and ODT and therefore take no special consideration of the regularity of results in CVT and ODT optimisation.

4.1. Initialisation for Constant Density

The Hammersley sequences used in this work are constructed using Eq. 4,

$$p = (\tfrac{i}{N}, \psi_2(i)) \in \mathbb{R}^2,$$

in 2D and

$$p = (\tfrac{i}{N}, \psi_2(i), \psi_3(i)) \in \mathbb{R}^3$$

in 3D, for $i = 0, \ldots, N-1$ and N points. To generate points in the Hammersley sequence, we use the simple algorithm described by Wong et al. [28] which requires approximately $log_2(j)$ bitwise shifts, multiplications and additions. This algorithm focuses on generating the Hammersley sequence in the plane and the sphere, and is therefore extended slightly following [29] to allow us to also sample in 3D. We use the prime bases $b_1 = 2$ and $b_2 = 3$ in order to generate samples with the most uniformly distributed pattern, and shift i/N by 0.5 to centre the sequence [28]. To generate pseudorandom numbers, the C++ Standard Library random number generator is used.

As an example of the sequence construction, we present the case of generating $N = 5$ points in $\mathbb{R}^2$ in base $b = 2$. Using Eq. 3, Table 1 shows the sequence expansion. Column $\mathbf{a_i}$ shows the integer input for i and its binary expansion. Columns 2–4 show the expansion of the digits. Column 5 shows the x value of the co-ordinate, and 6, the y value; the summation of columns 2–4.

$\mathbf{a_i}$	$\mathbf{a_{i2}2^{-2-1}}$	$\mathbf{a_{i1}2^{-1-1}}$	$\mathbf{a_{i0}2^{-0-1}}$	$\mathbf{(i+0.5)/N}$	$\boldsymbol{\psi_2(i)}$
$0 \rightarrow (000)$	0	0	0	0.1	0
$1 \rightarrow (001)$	0	0	0.5	0.3	0.5
$2 \rightarrow (010)$	0	0.25	0	0.5	0.25
$3 \rightarrow (011)$	0	0.25	0.5	0.7	0.75
$4 \rightarrow (100)$	0.125	0	0	0.9	0.125

Table 1: Generation of 2D Hammersley point set for $N = 5$ and $b = 2$. The final two colums represent x and y positions for a point.

4.2. Density Controlled CVT Initialisation

For a non-constant density function δ in Eq. 1 the initialisation for CVT generation has to be done more carefully. Du and Wang [30] show that the energy of a seed $\mathbf{x}$ in a CVT is inversely proportional to the density function, i.e.,

$$\int_{V_i} \|\mathbf{x}\|^2 d\sigma \propto (1/\delta)^{d+2/d},$$

for a d-dimensional Voronoi cell V_i. So initialising the optimisation with respect to this relation is advantageous because it avoids long-distance migration of sites during the optimisation process, which would slow down convergence and potentially inhibit any improvement over the random initialisation in the final distribution. To adjust the Hammersley and pseudorandom sequences, we introduce a function which maps a set of uniformly distributed sites $\mathbf{X}$ to a set $\mathbf{X}'$ of sites that are distributed according to the density function δ. This function should ideally preserve the geometric properties of $\mathbf{X}$, such that we can still improve the CVT by using the Hammersley sequence as an initialisation.

One approach is to use an error diffusion strategy [31]. However, this method produces an initial distribution which only approximately follows the prescribed density. We improve upon this by implementing a discrete approximation of the inverse density (if an analytic expression is not given) in order to initialise a CVT with respect to δ: $\mathbf{X}' = CDF_\delta(\mathbf{X})$, where CDF represents the cumulative density function. As δ is defined on the whole d-dimensional region, we integrate over each independent component of this joint function to get a piecewise quadratic approximation of the CDF for each component of the joint δ. Then for each coordinate of the uniformly distributed site $\mathbf{x}_i$, we compute the previous and next values, a_i and b_i, of the respective CDF along the axis, bounding the co-ordinate of $\mathbf{x}_i$ on that axis. We then solve the quadratic bounded by a_i and b_i for the co-ordinate of $\mathbf{x}_i$, resulting in a new co-ordinate. Solving this for each component of $\mathbf{x}_i$ gives us a new site $\mathbf{x}'_i$ distributed with respect to δ. Doing this for the set of all uniformly distributed sites $\mathbf{X}$ gives us a set of sites $\mathbf{X}'$ distributed according to δ.

Currently, our method does not extend to density controlled initialisation for non-regular domains. A näive solution to this problem would be to compute the CDF in a bounding box of the domain, and reject samples not within this domain. This process may then be accelerated with a quad-tree data structure. However, we have not currently investigated this, and it may be considered as future work. In Section 5.4, we show results

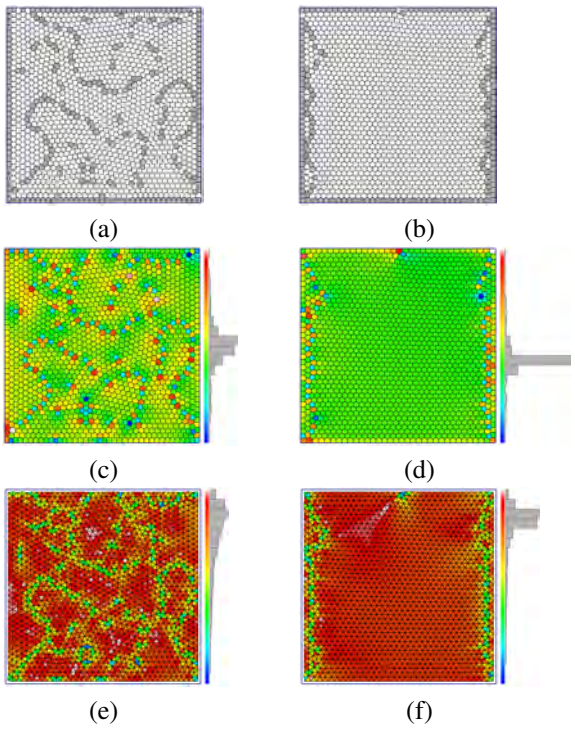

Figure 3: Two CVTs with 1,500 seeds in a square. Left column: random initialisation, converging to an energy of 1.29685×10^{-3}; right column: Hammersley initialisation, converging to an energy of 1.29328×10^{-3}.

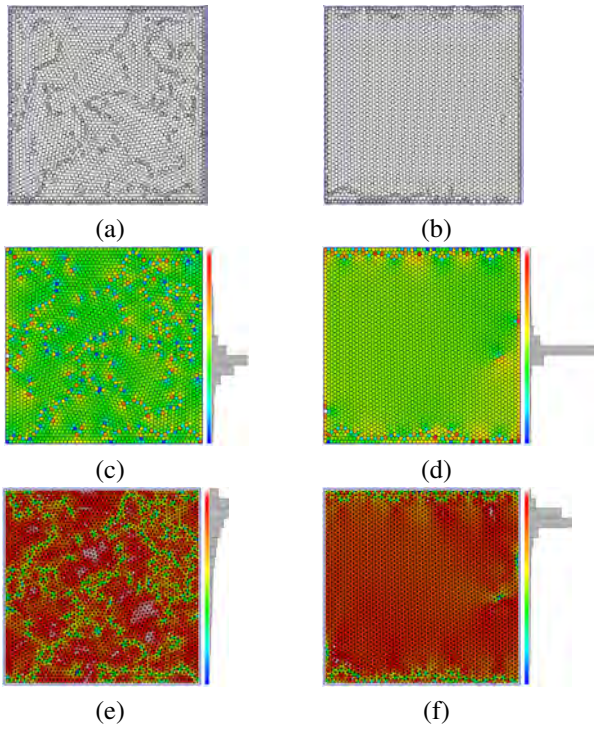

Figure 4: Two CVTs with 3,000 seeds in a square. Left column: random initialisation, converging to an energy of 6.47582×10^{-4}; right column: Hammersley initialisation, converging to an energy of 6.46110×10^{-4}.

for our inverse method in order to produce density-controlled CVTs.

5. Experimental Results

In this section, we compare CVT and ODT generation using pseudorandom and Hammersley initialisations. Other low-discrepancy sequences performed well, too, but Hammersley was consistently better and hence these results are omitted. Since the criteria to measure the quality of the results are different in 2D and 3D domains, we categorise the experiments accordingly.

In 2D we investigate three properties of the CVTs and ODTs, focusing mainly on CVTs. Firstly, the regularity of Voronoi regions, or the regularity of the vertices in the dual triangle mesh, is highly important in computer graphics [32]. We assess this regularity by drawing the Voronoi tessellation of the output sites and by highlighting any non-six-sided Voronoi cell in grey (denoted (a) and (b) in the figures with pseudorandom and Hammersley initialisation respectively). Fewer than six edges implies a non-regular tessellation, which is undesirable. According to Gersho's conjecture [33], asymptotically the energy of each seed within a CVT should be the same in 2D. Therefore, secondly, we show the energy value of each site (denoted (c) and (d) in the figures): high energy is shown in white and low energy in blue, along with a histogram representing the proportion of the cells with that energy. A uniform cell energy across the tessellation is desirable. Thirdly, the smallest angle, measured for each triangle in the dual triangulation of the Voronoi tessellation, is also important in computer graphics and finite element analysis (denoted (e) and (f) in the figures): a large smallest angle is shown in white and a small minimum angle in blue. The larger the smallest angle the better the triangulation; optimally each triangle should be equilateral and therefore the smallest angle within a triangle would be 60°. In addition we show the graph of the CVT function value against the number of iterations. Note that the cell energy is not shown for ODT and thus the smallest angle is denoted by (c) and (d) in these figures.

In 3D, even in infinite space, there is no widely accepted *optimal* tetrahedral partition (as dual of the Voronoi tessellation). Therefore, we do not compare the regularity between the different initialisations. However, the dihedral angles (the angle between two planes) of the tetrahedrons are important to evaluate tetrahedral mesh quality. A tetrahedron with a dihedral angle close to 0 or π is often called a *sliver* [4] and can lead to a matrix with a large condition number in numerical simulation. This in turn indicates numerical instability and is therefore undesirable.

For each figure (both CVT and ODT), results were run multiple times for the pseudorandom distribution, and a typical example is shown. In addition, the stopping criteria is set to $\|g\|/\|\mathbf{X}\| < 1 \times 10^{-10}$ in all examples, where g is the gradient, and $\mathbf{X}$ the variable (vector of seed point coordinates).

5.1. 2D CVT and ODT Examples

We first show the 2D CVT and ODT generation results, starting with a square domain with number of sites gradually in-

creasing from 800 to 10,000. Fig. 1 shows the CVT in a square with 800 sites. The top row of images shows results for a random initialisation and the bottom row for the Hammersley initialisation. Random initialisation leads to many non-hexagonal Voronoi cells, spread throughout the region. Using the Hammersley initialisation significantly improves the regularity of the Voronoi cells, showing significantly fewer irregular cells toward the centre of the region with most of the irregular cells located along the boundary. Moreover, the energy per cell statistic shows that each cell has a more uniform energy distribution for the Hammersley initialisation compared to the random initialisation. In addition, the triangulation (the dual of the Voronoi tessellation) has only a few triangles with a small smallest angle for the Hammersley initialisation. In comparison, using a random initialisation results in a very inconsistent triangulation with a large quantity of the triangles with a small smallest angle.

The result of Hammersley initialisation is consistently better in all the experiments on the square region as shown in Figs. 3, 4, 5, 6. In these figures the left column shows results for a random initialisation and the right column for the Hammersley sequence. We also include several examples for some general regions, including the hexagon in Fig. 7, the flower in Fig. 8, the butterfly in Fig. 9 and the cross in Fig. 10.

Fig. 11 shows the CVT function value against the number of P-L-BFGS iterations for random initialisation and Hammersley initialisation for 2300 sites within the butterfly region (see Fig.9). We show only this example, as the results are very similar for all other shapes tested. The Hammersley initialisation starts with a much lower energy value than random initialisation and reaches a value very close to the converged value after fewer than 10 iterations. The random initialisation does not reach this function value even after several hundred iterations. The pseudorandom case was initialised three times, and the average function value is shown.

Fig. 12 shows results for ODT using 800 sites in a square, showing the regularity of the Voronoi regions and the smallest angle measurement. The left column shows results for a random initialisation and the right column for the Hammersley initialisation. Using the Hammersley initialisation improves the result considerably, but each result is worse than that for the CVT counterpart (see Fig. 1). This is largely due to the less optimal algorithm used for computation of the ODT.

5.2. 2D CVT Generation Timing

Next we discuss timing results for CVT generation in 2D with the random and Hammersley initialisation. For all tests we generate initialisation sites in the unit square with uniform density. We run the P-L-BFGS method terminating the optimisation process when the function value drops below 1×10^{-4} for 10^3 samples, 1×10^{-6} for 10^4 samples, and 1×10^{-7} for 10^5 samples. For each point set, we run the initialisation and optimisation process ten times and record the mean computation time. Whilst the Hammersley sequence is deterministic, the optimisation process is not. Thus, results may vary slightly.

Table 2 lists the timing results for this experiment. These show that when CVT is initialised with Hammersley instead of pseudorandom point samples, a large reduction in the time required for the P-L-BFGS optimisation process to reach the fixed

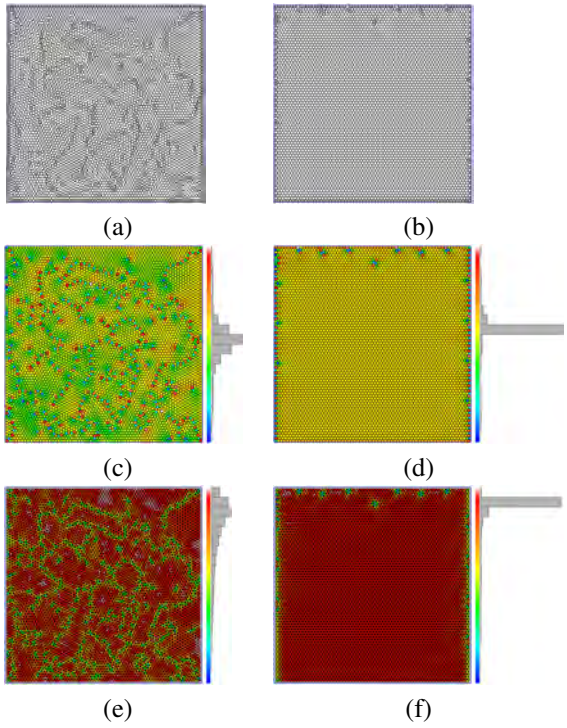

Figure 5: Two CVTs with 5,000 seeds in a square. Left column: random initialisation, converging to an energy of 3.88301×10^{-4}; right column: Hammersley initialisation, converging to an energy of 3.86539×10^{-4}.

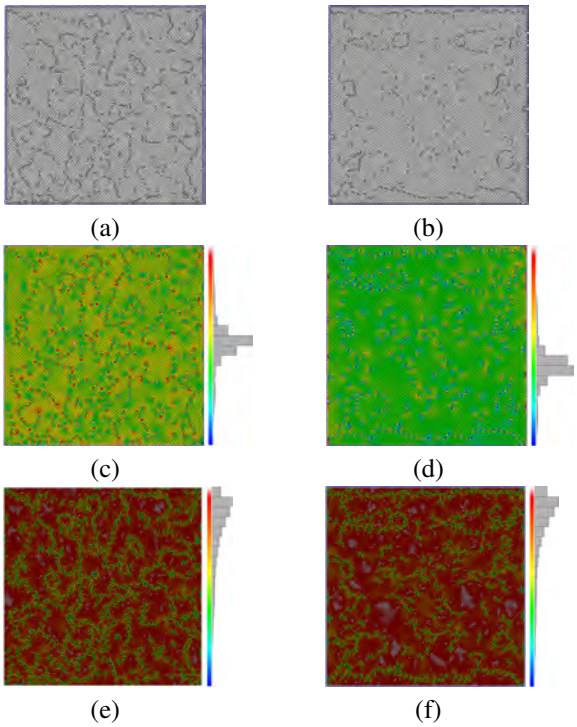

Figure 6: Two CVTs with 10,000 seeds in a square. Left column: a random initialisation, converging to an energy of 1.94032×10^{-4}; right column: the Hammersley initialisation, converging to an energy of 1.93726×10^{-4}.

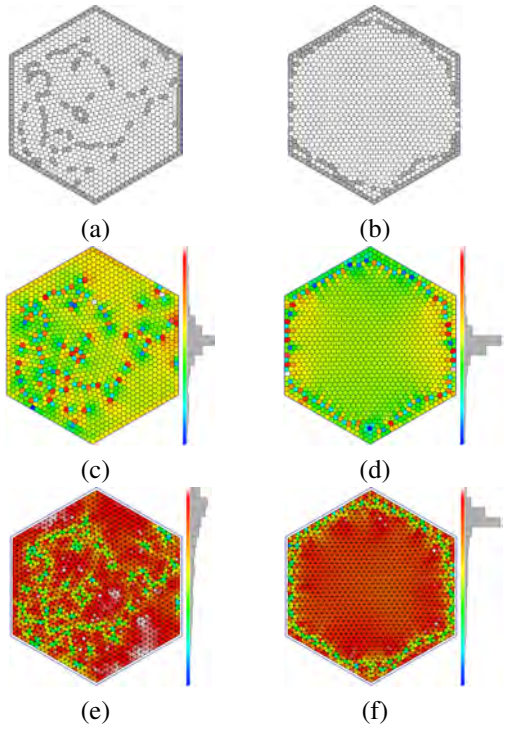

Figure 7: Two CVTs with $1,150$ seeds in a regular hexagon. Left column: random initialisation converging to an energy of 1.14076×10^2; right column: Hammersley initialisation converging to 1.14064×10^2.

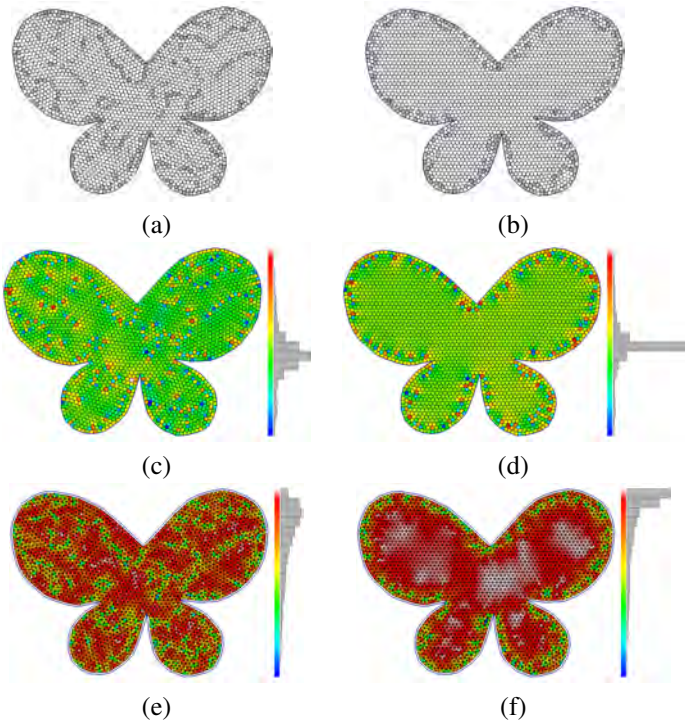

Figure 9: Two CVTs with $2,300$ seeds in a butterfly. Left column: random initialisation converging to an energy of 2.20264×10^{-2}; right column: Hammersley initialisation converging to 2.19553×10^{-2}.

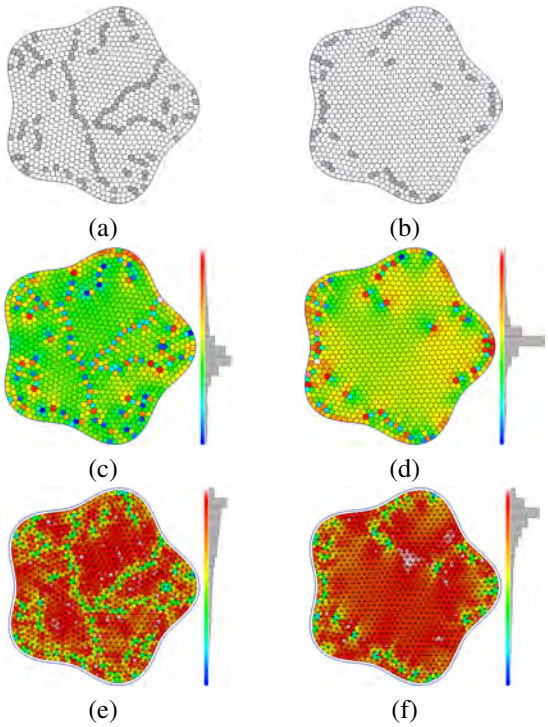

Figure 8: Two CVTs with $1,000$ seeds inside a flower shape. Left column: random initialisation converging to an energy of 1.93848×10^2; right column: Hammersley initialisation converging to 1.92862×10^2.

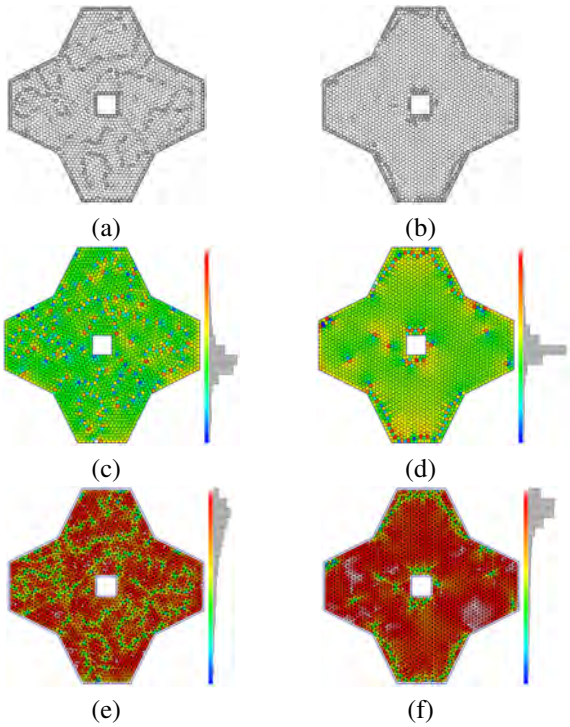

Figure 10: Two CVTs with $2,000$ seeds in a cross. Left column: random initialisation converging to an energy of 5.88808×10^{-3}; right column: Hammersley initialisation converging to5.86236×10^{-3}.

energy value. Using the Hammersley initialisation reduces the total computation time by 49% for $1,000$ samples, and 45% for $10,000$ and $100,000$ samples. In addition, the number of energy function evaluations is reduced by 55% for $1,000$ sam-

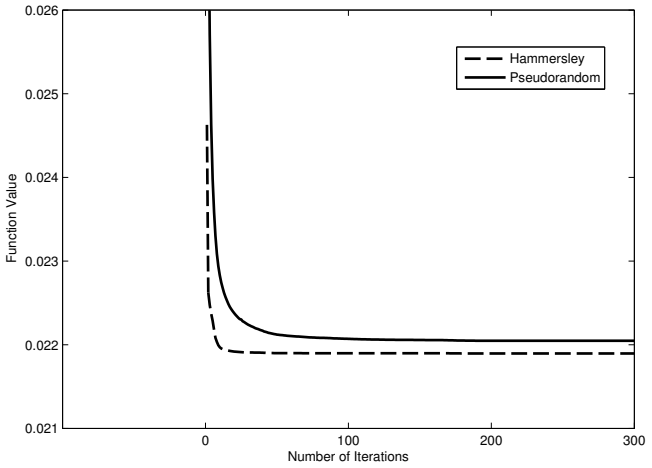

Figure 11: The CVT function value plotted over the number of iterations for 2300 sites in the butterfly using pseudorandom and Hammersley initialisation.

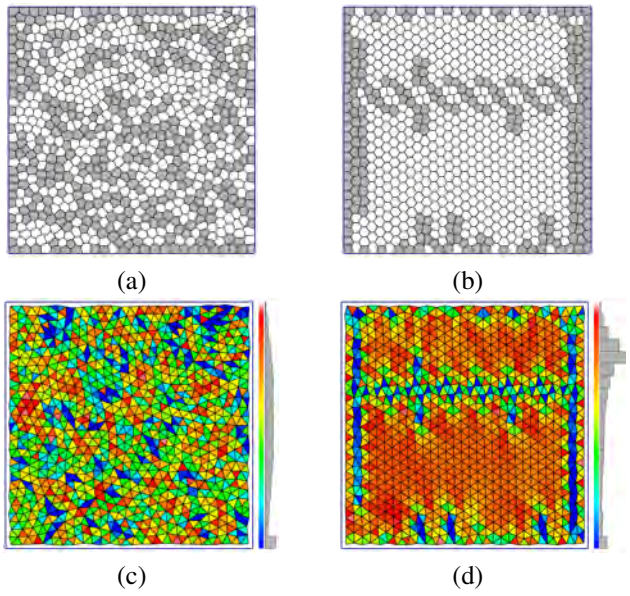

Figure 12: Two ODTs with 800 seeds in a square. Left column: random initialisation converging to an energy of 4.65419×10^{-4}; right column: Hammersley initialisation converging to 4.52950×10^{-4}.

ples, and 44% for 10,000 and 100,000 samples. This significant speed increase is largely due to the lower energy value achieved by the Hammersley initialisation.

For small point sample sets, e.g. 1,000, random points are actually more expensive to generate, per-point, than Hammersley points. This is due to the cost of generating the seeding object from which the samples are generated. For larger sample sizes, reaching 100,000, this overhead is no longer noticeable and a random sample takes approximately 7.2 nano-seconds to generate, compared to approximately 8.5 nano-seconds for a Hammersley sample.

5.3. Volumetric CVTs

For volumetric CVTs we see similar improvements to the 2D case when using the Hammersley initialisation. In a cube the Hammersley initialisation with 2,000 sites reduces the number of slivers in the dual tetrahedrisation (see Fig. 13). The Hammersley initialisation results in 10 slivers with dihedral angle less than 10°, compared with 15 for the random initialisation. In addition, the number of slivers with dihedral angle less than 15° for the Hammersley sequence is 20, compared with 21 for the random initialisation. Fig. 14 shows results for the Fandisk model, sampled with 10,000 sites. The Hammersley initialisation has 96 slivers with dihedral angle less than 10°, compared with 118 for the random initialisation. Also, the number of slivers with dihedral angle less than 15° for the Hammersley initialisation is 181, compared with 183 for the random initialisation.

Sites	Method	Init	Optim.	Total	#Func. Eval
1,000	Random.	0.000130	0.5843	0.5844	31.3
	Hamm.	0.000087	0.2951	0.2952	14.0
10,000	Random.	0.00080	18.6304	18.6312	165.3
	Hamm.	0.00086	10.1640	10.1649	93.2
100,000	Random.	0.00723	186.8009	186.8081	187.4
	Hamm.	0.00853	101.8421	101.8506	104.4

Table 2: Mean timing results in seconds for initialisation, CVT optimisation, and total time, and the mean number of function evaluations using random and Hammersley initialisation.

5.4. Density-controlled CVTs

Fig. 15 shows results for CVT computation using the density function $x^4 + y^4 + 0.001$ with random and Hammersley initialisation. The inverse method introduced in Section 4.2 is used to adjust the density of the random and Hammersley initialisations. The Hammersley initialisation shows improved results when compared to the random initialisation and the tessellation appears to be more natural. However, these gains are quite marginal compared to the improvements shown for uniform density cases.

5.5. Summary

In summary, in almost all cases the Hammersley initialisation achieved better results than the random initialisation. This is particularly true for generating CVTs in 2D with uniform density where the Hammersley initialisation has the clear advantage of creating an overall considerably more regular tessellation with more evenly distributed energy. As, in general, the Hammersley initialisation starts with a smaller function value, we see an almost 50% reduction in the the total optimisation time and the number of function calls required to reach a specific energy threshold. In addition, a smaller energy value and hence, a more regular triangulation can be achieved. While there is also a clear improvement for ODT generation, it seems that the distributions could be further improved. This is likely a result of less well-developed algorithms for ODT generation. For volumetric CVTs we obtained results similar to the 2D CVT case, even if the effect is less prominent. For density controlled CVTs only a small improvement could be achieved; better results may be obtained by improving the computation of the in-

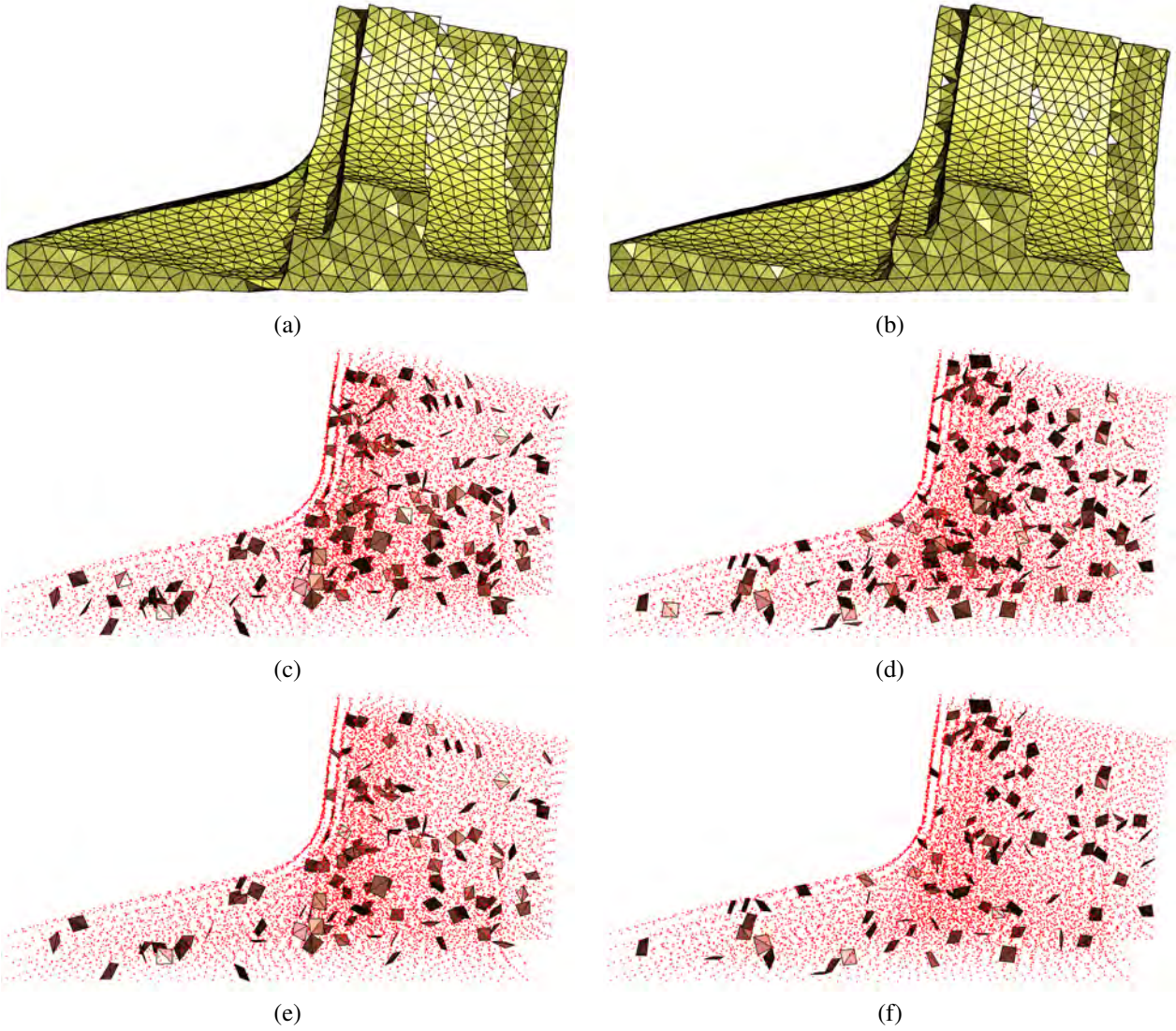

Figure 14: Two CVTs with 10,000 seeds in the Fandisk. Left column: (a) random initialisation converging to an energy of 1.96159 with 183 slivers (< 15°) in (c) and 118 slivers (< 10°) in (e); right column: (b) Hammersley initialisation converging to an energy of 1.95919 with 181 slivers (< 15°) in (d) and 96 slivers (< 10°) in (f).

verse of the density function in order to better preserve the geometric properties of the Hammersley sequence.

6. Conclusion

We have considered the use of different distributions of sites to initialise algorithms for generating CVTs and ODTs. In particular, the Hammersley low-discrepancy initialisation yields improved results compared to (pseudo-)random initialisation and other low-discrepancy sequences. The results are more regular and the total computation time is reduced by almost 50%. In addition, the Hammersley sequence is essentially as quick to compute as random samples, avoiding any computational overhead in the initialisation. It has similar regularity to regular grid initialisations, but can be easier adjusted to domains with complicated boundaries.

Overall, the experimental results show that CVT and ODT clearly depend significantly on how the algorithms are initialised. Large improvements can be achieved by finding a better initialisation, which is not necessarily expensive to compute. For CVT and ODT generation with uniform density a clear improvement in the resulting distribution could be demonstrated by using the Hammersley sequence. For non-uniform densities the effect is marginal and improved initialisations may be found by improving the inverse density computation in order to preserve the geometric properties of the Hammersley sequence better [34]. In future work, improvements for non-uniform tessellations and an extension to higher-dimensional domains may be achieved by better understanding the properties of the initialisation that help to improve the regularity of the results and the "landscape" of the energy functions for ODT and CVT. In addition to this, we intend to pursue this initialisation approach to consider CVT generation on surfaces [35], though we believe that the Hammersley sequence may be too sensitive to the

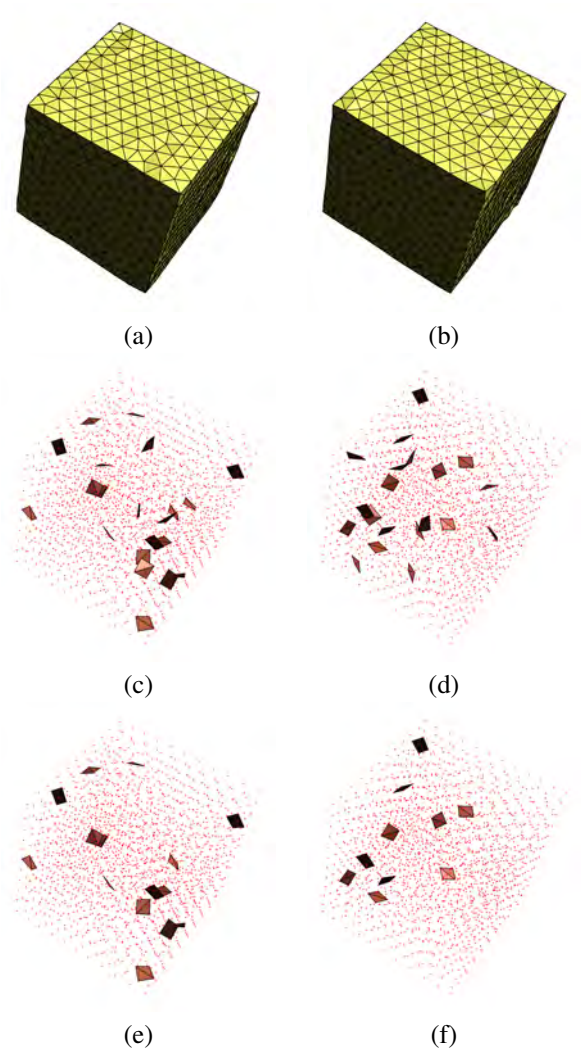

Figure 13: Two CVTs with 2,000 seeds in a cube. Left column: (a) random initialisation converging to an energy of 4.84254×10^{-2} with 21 slivers ($< 15°$) in (c) and 15 slivers ($< 10°$) in (e); right column: (b) Hammersley initialisation converging to an energy of 4.83987×10^{-3} with 20 slivers ($< 15°$) in (d) and 10 slivers ($< 10°$) in (f).

distortion introduced by a standard parameterisation method.

References

[1] Q. Du, V. Faber, M. Gunzburger, Centroidal Voronoi tessellations: applications and algorithms, SIAM Review 41 (4) (1999) 637–676.

[2] L. Chen, J. Xu, Optimal Delaunay triangulations, J. Computational Mathematics 22 (2) (2004) 299–308.

[3] P. M. Gruber, Optimum quantization and its applications, Advances in Mathematics 186 (2) (2004) 456–497.

[4] J. R. Shewchuk, What is a good linear element? interpolation, conditioning, and quality measures, in: Proc. 11th Int. Meshing Roundtable, 2002, pp. 115–126.

[5] P. Alliez, D. Cohen-Steiner, M. Yvinec, M. Desbrun, Variational tetrahedral meshing, ACM Trans. Graphics 24 (3) (2005) 617–625.

[6] V. Romero, J. Burkardt, M. Gunzburger, J. Peterson, Comparison of pure and "latinized" centroidal Voronoi tessellation against various other statistical sampling methods, J. Reliability Engineering and System Safety 91 (10–11) (2006) 1266–1280.

[7] D. R. Mitchell, Generating antialiased images at low sampling densities, in: M. C. Stone (Ed.), SIGGRAPH, 1987, pp. 65–72.

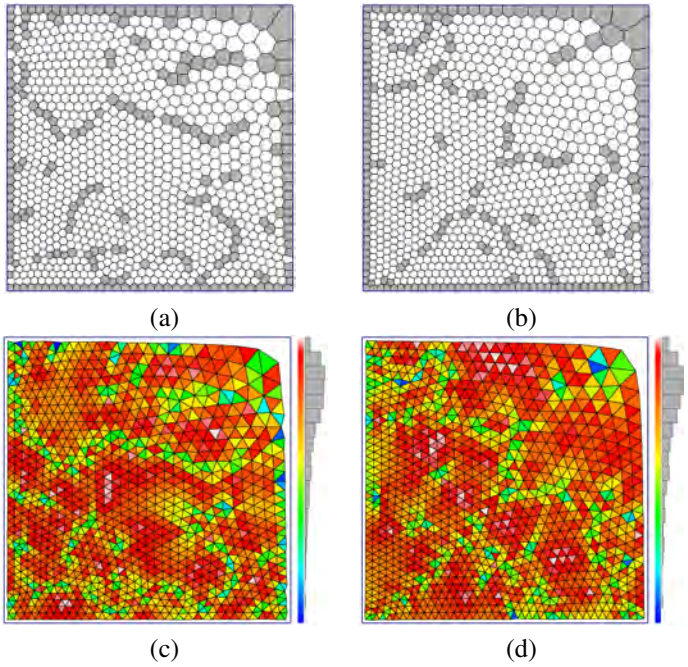

Figure 15: Two CVTs with 1,000 seeds in $[-1, 1]^2$ with density function $x^4 + y^4 + 0.001$. Left column: random initialisation converging to an energy of 6.09553×10^{-3}; right column: Hammersley initialisation converging to an energy of $6.08346e \times 10^{-3}$.

[8] Y. Liu, W. Wang, B. Lévy, F. Sun, D.-M. Yan, L. Lu, C. Yang, On centroidal Voronoi tessellation – energy smoothness and fast computation, ACM Trans. Graphics 28 (2009) 101:1–101:17.

[9] S. P. Lloyd, Least squares quantization in PCM, IEEE Trans. Information Theory 28 (2) (1982) 129–137.

[10] M. Moriguchi, K. Sugihara, A new initialization method for constructing centroidal Voronoi tessellations on surface meshes, in: Proc. Int. Symp. Voronoi Diagrams in Science and Engineering, 2006, pp. 159–165.

[11] H. Niederreiter, Random Number Generation and Quasi-Monte Carlo Methods, SIAM, 1992.

[12] J. Quinn, Low-discrepancy point sampling of 2d manifolds for visual computing, Ph.D. thesis, Cardiff University (2009).

[13] Q. Du, M. Gunzburger, L. Ju, Advances in studies and applications of centroidal Voronoi tessellations, Numer. Math. Theory Methods Appl. 3 (2) (2010) 119–142.

[14] Q. Du, M. Emelianenko, Acceleration schemes for computing centroidal Voronoi tessellations, Numerical Linear Algebra with Applications 13 (2–3) (2006) 173–192.

[15] B. Cipra, What's Happening in the Mathematical Sciences (1995–1996), Vol. 3, Amer. Math. Soc., Providence, RI, 1996.

[16] P. Shirley, Discrepancy as a quality measure for sample distributions, in: Eurographics, Elsevier Science, 1991, pp. 183–94.

[17] J. A. Quinn, F. C. Langbein, R. R. Martin, Low-discrepancy point sampling of meshes for rendering, in: M. Botsch, R. Pajarola, B. Chen, M. Zwicker (Eds.), Symp. Point Based Graphics, 2007, pp. 19–28.

[18] X. Li, W. Wang, R. R. Martin, A. Bowyer, Using low-discrepancy sequences and the crofton formula to compute surface areas of geometric models., Computer-Aided Design 35 (9) (2003) 771–782.

[19] T. J. G. Davies, R. R. Martin, Low-discrepancy sequences for volume properties in solid modelling, in: Proc. CSG, 1998.

[20] F. J. Hickernell, A generalized discrepancy and quadrature error bound, Mathematics of Computation 67 (221) (1998) 299–322.

[21] D. P. Dobkin, D. Eppstein, D. P. Mitchell, Computing the discrepancy with applications to supersampling patterns, ACM Trans. Graphics 15 (4) (1996) 354–376.

[22] J. M. Hammersley, Monte Carlo methods for solving multivariable problems, Ann. New York Acad. Sci. 86 (1960) 844–874.

[23] J. H. Halton, On the efficiency of certain quasi-random sequences of

points in evaluating multi-dimensional integrals, Numer. Math. 2 (2) (1960) 84–90.
[24] J. G. V. D. Corput, Verteilungsfunktionen i, Nederl. Akad. Wetensch. Proc. Ser. B 38 (38) (1935) 813–821.
[25] M. Haahr, Random.org True Random Number Service, `http://www.random.org`.
[26] D. Arthur, S. Vassilvitskii, k-means++: the advantages of careful seeding, in: Proc. 18th ACM-SIAM Symp. Discrete Algorithms, 2007, pp. 1027–1035.
[27] A. Meyerson, Online facility location, in: Proc. 42nd IEEE Symp. Foundations of Computer Science, 2001, pp. 426–.
[28] T.-T. Wong, W.-S. Luk, P.-A. Heng, Sampling with Hammersley and Halton points, J. Graphics Tools 2 (2) (1997) 9–24.
[29] J. H. Halton, Algorithm 247: Radical-inverse quasi-random point sequence, Communications of the ACM 7 (12) (1964) 701–702.
[30] Q. Du, D. Wang, Tetrahedral mesh generation and optimization based on centroidal Voronoi tessellations, Int. J. Numerical Methods in Engineering 56 (9) (2003) 1355–1373.
[31] V. Ostromoukhov, A simple and efficient error-diffusion algorithm, in: Proc. 28th Conf. Computer Graphics and Interactive Techniques, 2001, pp. 567–572.
[32] Y. Li, E. Zhang, Y. Kobayashi, P. Wonka, Editing operations for irregular vertices in triangle meshes, ACM Trans. Graphics 29 (2010) 153:1–153:12.
[33] A. Gersho, Asymptotically optimal block quantization, IEEE Trans. Information Theory 25 (4) (1979) 373–380.
[34] J. Hartinger, R. Kainhofer, Non-uniform low-discrepancy sequence generation and integration of singular integrands, in: H. Niederreiter, D. Talay (Eds.), Monte Carlo and Quasi-Monte Carlo Methods, Springer, 2006.
[35] Q. Du, M. D. Gunzburger, L. Ju, Constrained centroidal Voronoi tesselations for surfaces, SIAM J. Sci. Comput. 24 (5) (2003) 1488–1506.