Fast Low-discrepancy Sampling of Parametric Surfaces and Meshes

Jonathan A. Quinn, Frank C. Langbein, Yu-Kun Lai, Ralph R. Martin

School of Computer Science & Informatics,
Cardiff University, UK
{j.a.quinn,f.c.langbein,yukun.lai,ralph}@cs.cf.ac.uk

Abstract. This paper summarises work on low-discrepancy sampling of parametric surfaces and meshes based on use of space-filling curves, and suggests two applications of such work: shape retrieval based on distance histograms, and non-photorealistic stroke-based rendering.

Keywords: Sampling, Low discrepancy, Parametric Surface, Mesh, Space-filling curve, Shape retrieval, Non-photorealistic rendering

1 Introduction

Sampling is a process that serves an important role in many fields of computer science and mathematics. Many applications use sampling to facilitate the discretisation of a continuous function, so that it can be represented and processed digitally. Alternatively, we often want to re-sample an existing discrete function, so that it may be stored in less memory, or is computationally cheaper to process. The most important requisite of sampling is that it allows us to reconstruct, or approximate, the original function as accurately as possible, given a fixed number of samples, *or*, uses the fewest samples to achieve a desired accuracy. Many functions have a non-uniform rate of change, in which case we may want to place more samples in regions where the function is rapidly changing, and fewer samples in regions where the function is flatter. Thus, in general, we may wish to prescribe a density on the function, and sample it according to the desired density. Often, this density may be a differentiable property of the function itself, such as curvature.

For a given number of samples, the relative amount of information present in a discretised function is dependent on the quality of the sampling. This leads us to the need to determine some measure of sampling quality. One way to do so is to compare the value of an integral to the numerical approximation of the same integral based on the samples, computed using a (quasi-)Monte Carlo method. *Discrepancy* [1] considers this error in terms of the supremum of the difference between the integral and the integral approximation, for finite subsets of the unit interval in 1D, or the unit square in 2D. A sampling with a high degree of equidistribution, with few holes, has a *low discrepancy*.

Another method to assess the quality of a sample distribution, popular for example in the computer graphics literature, is *spectral analysis*. Spectral analysis allows us to assess, on average in a distribution, the regularity of positions of samples in the local neighbourhood of a selected sample; it also allows us to assess anisotropy in the distribution. A Fourier transform is applied to the sampling, and its properties analysed in the frequency domain. Cook et al [2] argue that a *blue noise* distribution is desirable: a noisy spectrum with a deficiency of low-frequency energy, without any concentrated spikes. A blue noise sample distribution does not add its own structure to the function that it samples, yet the spacing between samples must be regular enough that for a given number of samples, the function is not undersampled in places. If samples were arranged on a grid, the distance between neighbours would clearly be highly regular, but this may lead to aliasing problems [3], and grid sampling requires that the number of samples be a perfect square. Samples not lying on a grid avoid the aliasing problem to varying degrees. Spectral analysis does not consider the equidistribution of the sampling, but is a good measure of how the spacing of the samples impacts the visual quality of the distribution, which is important in computer graphics.

In this paper, we look at the problem of sampling a 2-dimensional manifold embedded in 3-dimensional space, and applications. The problem of sampling such a manifold is that inevitably, it either must be performed within the embedding space, or in some parameterisation domain [4], a subset of $\mathbb{R}^2$. Sampling in the parameterisation domain is typically much more computationally efficient, but has the complication of typically requiring cutting or segmentation of any manifold with topology more complex than a disk. Additionally, care must be taken to avoid sampling artifacts at the boundaries between segments when reassembling the manifold. Sampling in the embedding space is computationally more expensive, but requires no cutting, parameterisation, or reassembly of the manifold, regardless of topology.

2D manifold sampling, especially the sampling of polygonal meshes, is an active area of research for both geometric processing and computer graphics. In the former, sampling is commonly used when performing decimation, such as the downsampling of large meshes captured using 3D scanning equipment. In the latter, sampling is often used to improve the output of anti-aliasing, physically-based rendering, and filtering algorithms. Four main approaches to sampling can be found in the literature, with a variety of construction methods for each; we now briefly describe them.

Random sampling is the random, or pseudo-random, distribution of samples. The distribution is one which, in the limit, results in a uniform probability distribution; i.e. there is an equal likelihood for a single sample point being placed at any position. However, uniformity only refers to the probability of each point individually, and hence, the points in a uniform, random, point distribution are uncorrelated. As a result, there is nothing to prevent random distributions from containing gaps without samples, and clumps of samples.

Quasi-random sequences provide a deterministic sampling approach with a high degree of equidistribution. The point locations in such sequences are correlated; the probability of a point being at some position is dependent on its position in the sequence. Correlation means that the quality of the whole point set is considered, rather than at each individual point, resulting in more even coverage of a domain, with fewer gaps and clumps. Quasi-random samples are highly structured, and are usually generated according to a lattice [1], or using variations of the van der Corput sequence [5, 6]. Often used as a replacement for the classic Monte Carlo method, they have been shown to reduce the number of samples needed to compute an integral to within a prescribed error bound, and maintain a consistently higher level of confidence in computations based upon them [7], i.e. errors are smaller and estimates of error are more likely to be accurate. The majority of quasi-random sequences provide optimal scaling of discrepancy with respect to the number of samples generated [1], and are thus often referred to as *low-discrepancy sequences*.

Poisson disk sample distributions are constructed based on a fixed minimum distance between neighbouring points. This structure can be achieved via a process of sampling and rejection [8, 9], or through optimisation methods that iteratively adjust sample positions [10, 11]. Optimisation methods approximate a *centroidal Voronoi tessellation* (CVT) [12], converging to a hexagonal grid away from boundaries. Optimisation is thus typically terminated before convergence, in order to avoid the presence of too much regularity. Rejection methods have limited regularity through the use of random sampling. However, both methods can result in a very regular *local* spacing of samples, and overall higher discrepancy than that produced by competing methods. Spectral analysis can show clear mid-frequency peaks, corresponding to a very strong regularity of local neighbourhood distances. A further drawback to Poisson disk methods is that if the sampling density function varies greatly, with adjacent very dense and very sparse regions, the computation of distances between neighbouring points, either for rejection of points, or for the computation of forces in optimisation, becomes very expensive.

Stratified sampling involves the subdivision of a domain into homogeneous subsets, or strata, within each of which a sample is placed. The position of the sample within a stratum may be random, or may be weighted toward the centre of the stratum. The approach was first introduced as a method of variance reduction for random sampling [1]. This degree of randomness, limited to the local stratum of a sample, avoids regularities in the point distribution prevalent in Poisson disk methods. [13] shows that for distributed ray-tracing, sampling patterns with very low discrepancy defined with respect to rectangular subsets (see [1]) may still cause artifacts. Stratified sampling, however, does not exhibit this problem, the output being rotationally invariant under discrepancy measures. Stratified sampling has the further benefit of demonstrating a discrepancy much lower than Poisson disk sampling.

We will later overview a series of related algorithms for stratified sampling of 2-manifolds embedded in 3D Euclidean space. The methods build upon the ideas

described in [14], and involve the sampling of space-filling curves, mapped to, or generated from, 2-manifolds. The space-filling curves act as a 1D stratification of the manifold, and allow us to replace two dimensional sampling by a one-dimensional sampling process in which we step along the curve, sampling a point when an integral on the curve reaches a certain threshold. This process is referred to as *generalised stratified surface sampling*.

We also later illustrate uses of generalised stratified surface sampling, with two applications: *shape retrieval* (see Section 5.2) and *stroke-based rendering* (see Section 5.1).

2 Discrepancy

In this section, we consider the discrepancy of a sample distribution, and how it may be computed in practice. We start by noting that, as the discrepancy $D(x)$ of a set P of point samples decreases, so does the approximation error for a Monte Carlo approximation $M(f, P)$ of a multivariate integral $I(f)$ of a function f [7]. Thus, the study of discrepancy of P is important, as it provides a quantitative measure for the quality of a set of samples P. Clearly is it is desirable to be able to generate sample sets with low discrepancy, and these have been employed in many different fields, such as computer graphics [15] and surface representation [16].

We can think of the discrepancy of a set as indicating the difference between the actual value of some integrand, and its approximation. More exactly [17]:

$$|I(f) - M(f, P)| \leq D(P)\mathrm{var}(f)$$

where $\mathrm{var}(f)$ is the variance of the integrand, defined as:

$$\mathrm{var}(f) = \int_{[0,1)^s} f(x)\ (I(f) - M(f, P))^2 d\lambda$$

and λ is the Lebesgue measure of a set. By lowering the discrepancy $D(P)$ of P, the approximation error, $|I(f) - M(f, P)|$, is reduced. The lower the discrepancy, the lower the difference between the continuous and approximated integrals.

In fact, various alternative definitions of discrepancy are used to measure the discrepancy of a sample set; the *star discrepancy*, D^*, is probably the most common [1]. Let $\mathcal{B}(w)$ with $w \in [0, 1)^s$ be an axis-aligned s-dimensional box with one corner at the origin, and P be the sample distribution in $[0, 1]^s$. The star discrepancy is the supremum of the magnitude of the difference between the cumulative distribution function and the volume:

$$D^*(P) = \sup_{w \in [0,1]^s} \left| \frac{|P \cap \mathcal{B}(w)|}{|P|} - \mathrm{vol}(\mathcal{B}(w)) \right|.$$

Thus, $D^*(P)$ is the supremum of the difference between the exact volume of an arbitrary subset and an approximation to it estimated by the number of samples

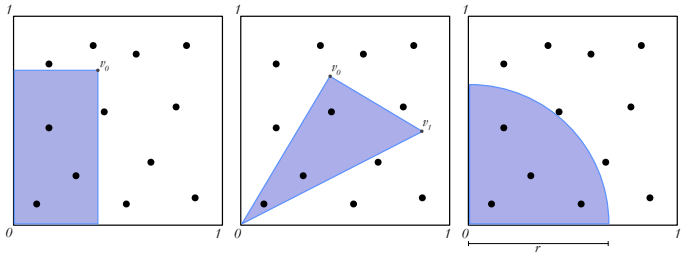

Fig. 1. Discrepancy subset test shapes: rectangle, triangle and quarter circle, with a corner at the origin.

inside it relative to the total number of samples, taken over all subsets which are axis aligned boxes in $[0,1)^s$ with one corner at the origin. A low-discrepancy point set has a low discrepancy for a fixed $|P|$. So we can define D_N^* as the optimally achievable $D^*(P)$ for all P with $|P| = N$, i.e. $D_N^* = \inf_{P \subset [0,1)^s; |P|=N} D^*(P)$. To assess a point distribution method, we are interested in the behaviour of the function $D^*(P)$ with respect to increasing $|P|$, as this determines the convergence properties of integral approximation.

For a uniform random sequence, as N increases, the discrepancy has scaling behaviour $O(N^{-1/2})$ [1], independently of the dimension s. Any sampling with better asymptotic behaviour will improve the convergence behaviour of the approximation. Achieving low discrepancy is difficult, but improving on a random distribution is desirable, and possible. The term low discrepancy is used to indicate an improved, rather than optimal, behaviour.

As noted earlier, one approach to reducing the variance of a sampling is to use stratified sampling. If an axis-aligned box $\mathcal{B}(w)$ is used to compute discrepancy, as indicated above, some quasi-random sequences asymptotically perform better than stratified sampling, both in theory and experimentally [18]. However, Dobkin et al [13] show that the asymptotic discrepancy behaviour of these methods is far worse when discrepancy is measured using other test shapes, such as non-axis-aligned rectangles; this observation is also confirmed in practice by Quinn et al [18] (see Fig 1). For applications such as supersampling in computer graphics, or mesh sampling in geometric processing, non-axis aligned, arbitrarily shaped regions are the norm, and in such cases quasi-random sequences are therefore less useful. Quinn et al show that stratified sampling produces better asymptotic behaviour when discrepancy is measured using test shapes defined by arbitrary edges in $[0,1]^2$. Stratified sampling also exhibits considerably better behaviour than Poisson disk sampling methods [19].

Later, we will use the concept of discrepancy as an experimental tool to analyse and compare various sampling methods. Whilst the testing domain may vary, including planar regions, parametric surfaces, and triangle meshes, we use a consistent general approach to numerical computation of discrepancy. For a specified domain and sampling shape, the generalised star discrepancy of a sam-

pling method is approximated by computing the maximum error for a finite number of test shapes, for sample sets of size $N = 2^l$ and $N = 2^l + 2^{l-1}$ for $l = 1, \ldots, 20$. This approach results in 40 sets with between 2 and 1572864 samples. Test shapes are generated until the estimated discrepancy for that value of N becomes stable. Results are averaged over three runs, and graphs are plotted as log(discrepancy) against $\log(N)$. A least squares line is fitted to the data for gradient computation, which lets us understand how discrepancy scales with size of sample point set. A steeper negative gradient implies accuracy improves more quickly with increasing number of samples, and hence the sampling approach has higher quality.

3 Spectral Analysis

In this section, we consider spectral analysis of sample distributions. Whilst discrepancy gives us a good measure of the uniformity of coverage of a distribution, it does not quantify the possible problems caused by any uniform structuring of a distribution, leading to visible quasi-regular patterns. It is generally accepted that for applications such as surface sampling for visualisation [20] and dithering [21], it is undesirable for a sample distribution to have a structure of its own, interfering with the existing structure or pattern of the object being rendered [21]. Distributions can be constructed, however, that are neither random (which could have large clumps or holes between points), nor quasi-structured, such as a jittering, or Poisson disc sampling.

Types of noise can be classified by their power spectrum, which gives the response at each frequency present. White noise for instance has a uniform power spectrum distribution, while blue noise, defined in the field of visual computing, avoids spikes, and has small low-frequency components. Constancy of medium to high frequencies means that there is little global density variation, and the absence of low-frequencies implies a regular local spacing between samples. Thus, a sample distribution with blue noise characteristics has a high visual quality. Regular grid sampling presents large, regular peaks due to aliasing. Random sampling results in white noise, which generally is undesirable if an image is being filtered or dithered [21].

An algorithm described in [21] can be used to compute the radially averaged power spectrum density (RAPSD) of a sample distribution. It builds upon Bartlett's approach [22] of computing the Fourier transform of a distribution, and averaging periodograms, which represent the spectral density of a signal. Averaging is done to reduce the variance of the plot, and is performed by computing the Fourier transform of subsets of the sample distribution, squaring their magnitudes, and dividing the total by the sample size. Ulichney [21] builds upon this in order to highlight the degree of radial symmetry in a distribution, by segmenting the distribution into concentric uniform-width rings, which are then averaged in a similar manner to Bartlett's original approach. An assumption is made that the distribution has a constant density, to justify the averaging.

Recent work on *differential domain analysis* [23], building upon [24, 8], allows for the detailed analysis of samplings with varying density and anisotropy, including those on mesh surfaces. Fourier spectral analysis relies on sample locations, and the authors demonstrate that this transform can be reformulated into an equivalent form that depends only on differentials of samples. This allows the method to work in different domains and with varying densities of samples. As the method is no longer reliant on distances, results are consistent between all surfaces, densities and anisotropy.

4 Surface Sampling

In this section, we first describe the core sampling approach utilised in this paper. We then summarise three previous algorithms that have been published that employ this approach; additional detail can be found in the respective papers. We start with a core description which covers the basic steps of the approach, which are common to the three algorithms.

Input. A 2-manifold M is provided as input, represented either as a parametric surface, or as a triangle mesh. Additionally, the user may supply a density function, ϕ which controls the local density of the sampling. For isotropic sampling, this may be expressed analytically, or given as a discrete, per-element density. For anisotropic sampling, the density is described as a tensor field defined on M, which again may be supplied in analytic form, or per element.

Output. The output of the sampling process is a set of sample points $p = \{p_0, \cdots, p_{N-1}\}$ that evenly covers the manifold M with a local density controlled by ϕ. In the anisotropic case, the typical spacing between points p depends on the direction across M, and is determined by the local tensor field.

Curve Generation. An approximation to a space-filling curve C is generated on the manifold. Space-filling curves are a continuous, surjective, self-intersecting mapping between $[0, 1]$ and $[0, 1]^d$. Geometric interpretations of these curves approximate the limit curves, and are usually self-avoiding; they pass no further than a certain maximum distance from every point of $[0, 1]^d$. Using space-filling curves here has the benefit of reducing a complex 2-manifold sampling problem to the more simple problem of sampling a 1D interval. An important property of the curve is that it should be spatially *coherent*, that is, points close on the curve should be close on the manifold (the reverse does not have to hold). Coherence avoids large changes in density at nearby points on the curve, assuming the density function varies smoothly; this avoids artifacts when sampling. If the manifold is parameterised in the unit square, a curve may be generated in the parameterisation domain and mapped onto the manifold. This is an efficient way to generate a curve on the manifold, but can lead to problems if the manifold has a high genus or uneven parametrisation, which we discuss further later. The output of curve generation is a piecewise linear curve $C = \{v_0, \cdots, v_{L-1}\} \in \mathbb{R}^3$

lying (approximately) on the manifold M.

Computing Surface Properties. Having generated a curve C on the manifold M, differential properties in the tangent plane of each curve vertex v_i are then computed. A local area γ_i, representing the area around the vertex, and a local density φ_i are then computed at v_i, the latter sampling the global density function ϕ. As noted, the local curvature at v_i is a common choice for the density function, but any other function over the manifold may also be used. φ is constant if a uniform distribution of samples is required. In the anisotropic case, the density is derived from the tensor field, which, in the anisotropic case, is aligned to the input tensor field.

Sampling. The per-vertex properties computed for the curve C lying on M are used to control sampling of C. Using a generated 1D sequence, samples are placed along C using a monotone non-decreasing cumulative density function ω_C which approximates the local area and surface density integral:

$$\omega_C(v_i) = \sum_{i=0}^{l-1} \gamma_i \varphi_i \approx \int_M \gamma \phi C dA$$

where l is the number of vertices of C. A set of N 1D samples is generated: $q = \{q_0, \cdots, q_N - 1\} \in [0, 1]$; they may be stratified, or generated deterministically within the interval $[0, 1]$. For each q_j, the algorithm then steps along the curve, and when $\omega_C(v_i)$ becomes larger than $q_j \omega_C(v_{l-1})$ (accounting for the total accumulated density), we place a sample point p_j at v_i. The output of this process is a set of points $p = \{p_0, \cdots, p_{N-1}\}$ lying on M.

We now examine in more detail how the above scheme can be made to work in three particular cases: isotropic sampling of parametric surfaces, isotropic sampling of triangle meshes, and anisotropic sampling of triangle meshes.

4.1 Parametric surfaces

In this section, we summarise the algorithm introduced in [18] for isotropic sampling of parametric surfaces, based on the general approach described above. The user supplies (i) a parametric surface $f : [0, 1]^2 \to \mathbb{R}^3$ in a manifold M_p, with the unit square as a normalised parameter domain, (ii) a non-negative bounded density function $\phi : S \to \mathbb{R}_0^+$, and (iii) a desired number of points N. The algorithm generates a set of points P equidistributed on M_p with respect to ϕ.

Mapping the parameter domain to a 2-manifold in 3D causes a non-uniform stretching of the domain. To counteract this, we generate an adaptive approximation to the space-filling curve, that places more vertices in areas of higher stretch, ensuring that the curve lies within a maximum distance of any point on the surface (see Fig. 2). An adaptive space-filling curve is generated using a tree-structure defined according to the type of space-filling curve, which facilitates further partitioning of the parameter domain $[0, 1]^2$. Adaption takes into

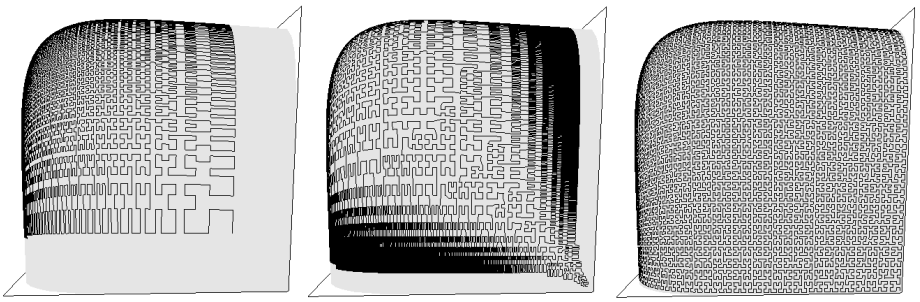

Fig. 2. Space-filling curve approximation on a surface with extreme parametric stretch. Left: uniform curve. Middle: adaptive curve. Right: re-parameterised curve.

account both areal stretch caused by parameterisation, and the desired sampling density, and is controlled by computing $\gamma_i \varphi_i$ for the surface patch covered by a subset of the curve, determining whether partitioning should continue.

Surface area and curvature are computed using the first and second fundamental forms, which are straightforward to derive from f. The first fundamental form defines metric properties on a surface such as angles and distances, whilst the second fundamental form combined with the first fundamental form allows us to compute curvature properties. If a parametric function f causes an extreme stretching of the domain, rather than adaptively generating the curve to a very deep level, a better alternative may be to reparameterise the surface with less parametric stretching (see Fig 2).

Various space-filling curves have been investigated for use in this algorithm, including the Hilbert, Peano and alternative-construction Peano curves [25, 3] (see Fig 3). The Hilbert cure was found to give best results for stratified sampling, in terms of spatial coherence and discrepancy [3].

We have also experimentally investigated the discrepancy of the resulting point sets. Computing the discrepancy on arbitrary surfaces is non-trivial, and we show results of sampling in the plane, and on the surface of a sphere. For the planar case, triangular, circular and rectangular sampling subset shapes were used (Fig 1), while on the sphere, spherical triangles were used. Results show that this method produces much lower discrepancy than random sampling, for samples produced on the plane, for all subset shapes. For the circular subset shape, the method performed as well as the best low-discrepancy sequences. For the triangular subset shape, the method outperformed other low-discrepancy methods. On the surface of a sphere, our method performed as well as known low-discrepancy methods specific to sampling the sphere. Fig 4 demonstrates these results, showing how discrepancy scales with the number of samples. Further, results are demonstrated for the discrepancy of this method where samples are generated with respect to a density function on the plane (Fig 5). For this approach, rather than simply computing the volume of the sample subset, we integrate over the function defined in the domain. The described method performed

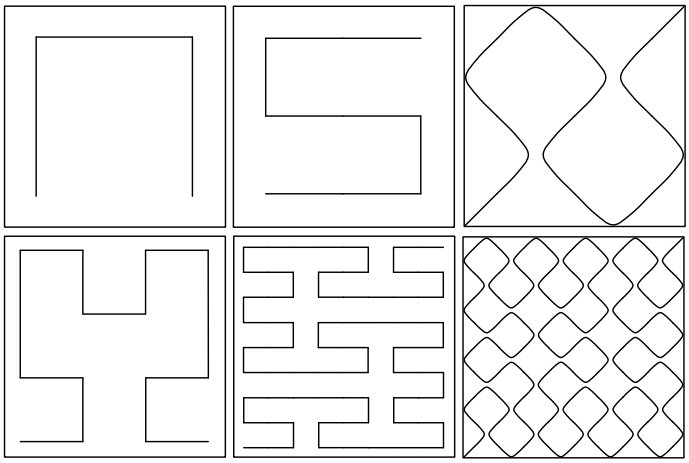

Fig. 3. Left to right: the first two iterations of the Hilbert curve, the Peano curve, and the variant Peano curve.

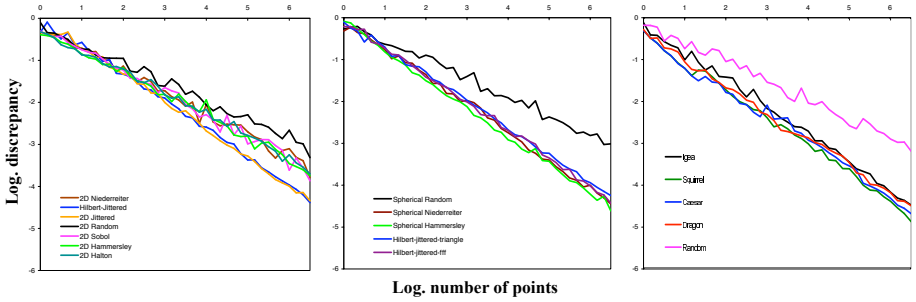

Fig. 4. Discrepancy. Left: on a plane, measured using quarter-circles, middle: on a parametric sphere, measured using spherical triangles, right: on various meshes.

consistently with other measures. Fig 6 shows visual results of this method for the eight surface.

4.2 Triangle meshes

In this section, we summarise the algorithm in [26] for isotropic sampling of triangle meshes. The user supplies a triangle mesh M_t. This algorithm computes topological cuts on the mesh, to ensure that the cut mesh M_c is homeomorphic to a disk, and can be simply parameterised in the plane. The algorithm then finds a conformal parameterisation $f : M_c \in [0,1]^2 \rightarrow M_{uv} \in \mathbb{R}^3$ for the mesh. A space-filling curve is then generated in $[0,1]^2$, and mapped to the original mesh M_t. It is then sampled, to produce a set of equidistributed points P on the mesh with respect to the density function ϕ (see Fig 7).

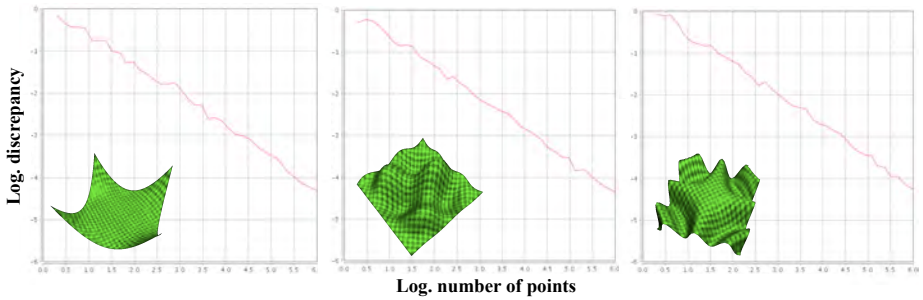

Fig. 5. Discrepancy in the plane, assessed using rectangular subsets, for various density functions: x^2y^2, $\cos 3x \sin 3y \cos xy + 1$, and $\cos 10xy + 1$.

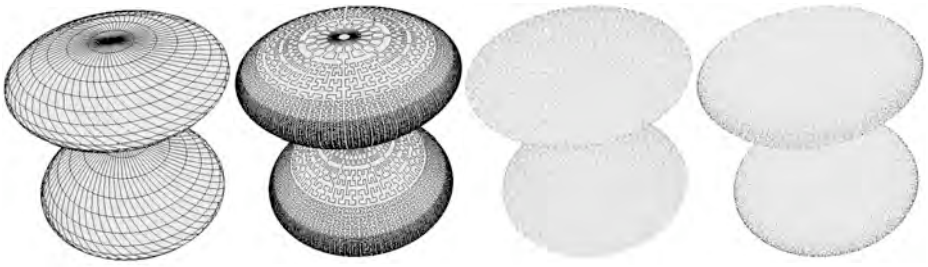

Fig. 6. Eight surface. Left to right: parametrisation, adaptive Hilbert curve, uniform sampling density, sampling density proportional to mean curvature.

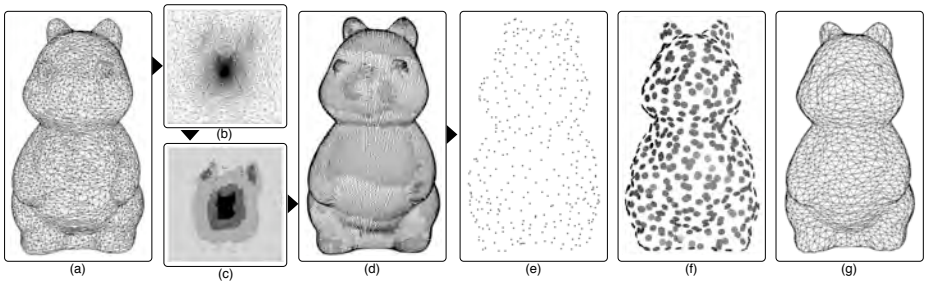

Fig. 7. Sampling a squirrel mesh: (a) original mesh, (b) parameterisation; (c) adaptive Hilbert curve in 2D, (d) adaptive Hilbert curve mapped to 3D, (e) uniform sampling, (f), (g) surfel primitive rendering with differing numbers of samples.

The input mesh may be of arbitrary topology, and thus the algorithm first cuts the mesh to a topological disk by seeking local, rather than global, extremities [27]. Methods have been described [28] that compute globally optimal cuts allowing a mesh to be parameterised with minimal areal and angular stretch. However, as our method can correct (at least to some degree) for area stretch caused by the parameterisation (see Fig 8), a less optimal, but less computa-

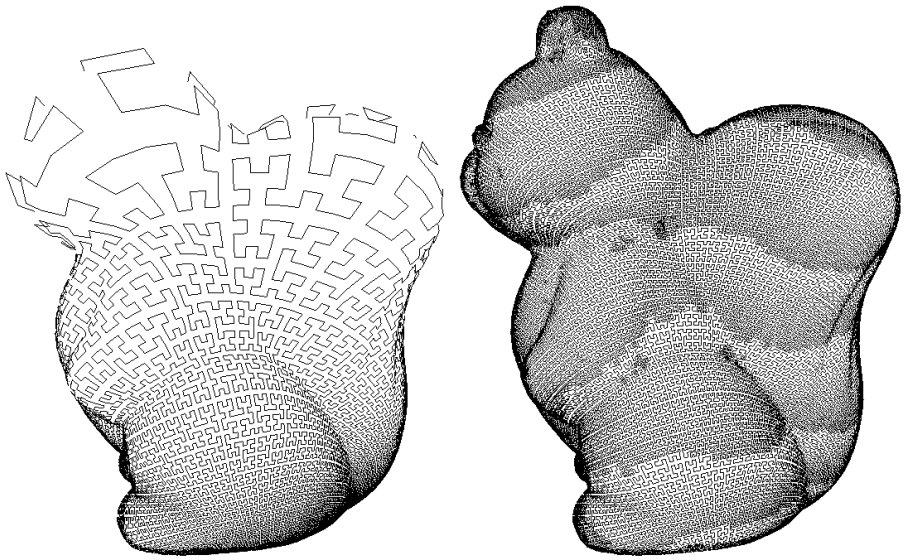

Fig. 8. Left: uniform, and right: adaptive Hilbert Curve.

tionally expensive, approach to cutting can be used. A fast, conformal, parameterisation f is applied to the cut mesh M_c in order to reduce angular stretch in the parameterised mesh.

A Hilbert curve in $[0,1]^2$ is then generated, which is mapped onto M_t. A GPU-based rasterisation method is used to compute the containing triangle for each vertex of the curve, and its barycentric coordinates are computed within that triangle. Using the associated triangle in M_t, the point is then converted back to Euclidean coordinates on the mesh surface. Local area and curvature for each curve vertex on M_t are computed using the methods described by Meyer et al [29].

To examine sampling quality, the discrepancy of samples on the mesh, and the Hausdorff distance between remeshed surfaces, are considered [30]. The Hausdorff distance is used to measure the approximation error between a surface and a resampling of the same surface. When computing discrepancy, test shapes are chosen to be random contiguous subsets of the mesh M_t. A seed vertex is chosen, and the test shape grown in rings to a random size. Results show similar discrepancy scaling benefits for the above approach over random sampling as were achieved in planar cases as described in Section 4.1 (again see Fig 4).

To assess sample quality using the Hausdorff distance, a mesh is computed from the sample points, and the accuracy of this remeshing, with respect to the original mesh, is computed; this is done for various degrees of decimation of the input meshes (see Fig 10). The Metro tool [30] is used to compute the Hausdorff distance between two meshes $M_{\mathcal{I}}$ and $M_{\mathcal{II}}$, using points on the respective

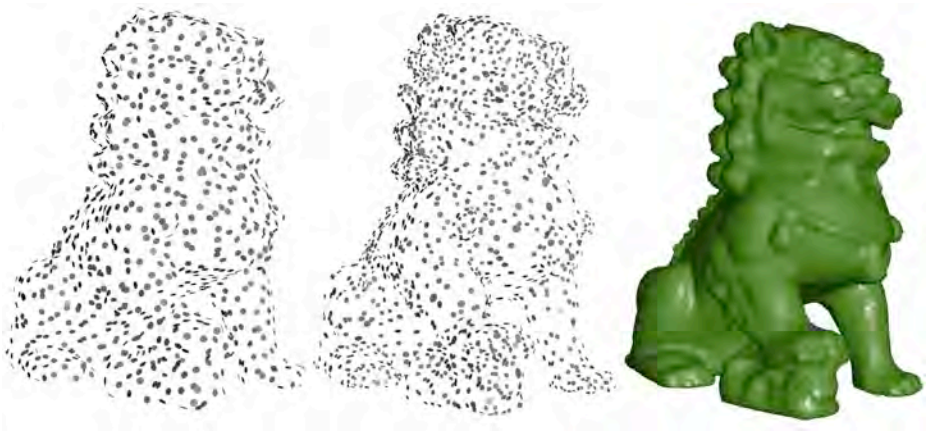

Fig. 9. Chinese lion mesh. Left to right: splatted uniformly, splatted with respect to mean curvature, and rendered.

surfaces, $p_{\mathcal{I}}$ and $p_{\mathcal{II}}$:

$$D_{avg}(M_{\mathcal{I}}, M_{\mathcal{II}}) = \frac{\int_{M_{\mathcal{I}}} \inf_{p_{\mathcal{II}} \in M_{\mathcal{II}}} d(p_{\mathcal{I}}, p_{\mathcal{II}})\, dA}{\text{area}(M_{\mathcal{I}})} \tag{1}$$

Computing the distance in both directions, and taking the maximum, gives us the symmetric Hausdorff distance. Results show that using our samples to produce a new mesh with 25% of the original triangles, yields a 0.02% normalised distance between the original mesh and the remeshed surface. Such accurate reproduction is competitive with leading remeshing methods [31], yet the mesh vertices produced by this approach also have a low discrepancy. Visual results are shown for splatting (see Fig 9) and remeshing (see Fig 10).

An application of the method is also described in [26], taking advantage of the very high speed with which resampling can be performed, to provide real-time fine-grain level of detail and view dependent rendering. In this case, rendering emphasises silhouette edges by controlling the number of samples, as verified by the views shown in Fig 11; the picture the viewer would see is on the left.

4.3 Anisotropic sampling of triangle meshes

In this section, we summarise the algorithm in [19] for anisotropic mesh sampling. The user supplies a triangle mesh M_A with an approximation of a smooth two dimensional tensor field, defined as a positive-definite, symmetric, rank 2 tensor T_i defined at each mesh vertex. The tensor is used to create an anisotropic space-filling curve, which is aligned with the field (rather than oriented to follow the arbitrary direction of the parametrisation as in previous approaches; see Fig 12). When sampled, this oriented curve produces an anisotropic sampling of M_A, where the spacing between neighbouring samples varies with direction.

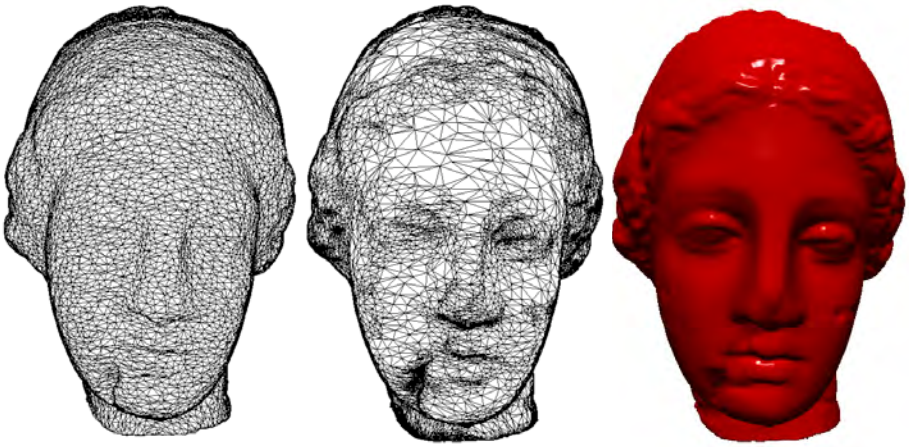

Fig. 10. Igea mesh. Left to right: remeshed uniformly, remeshed with respect to mean curvature, and rendered.

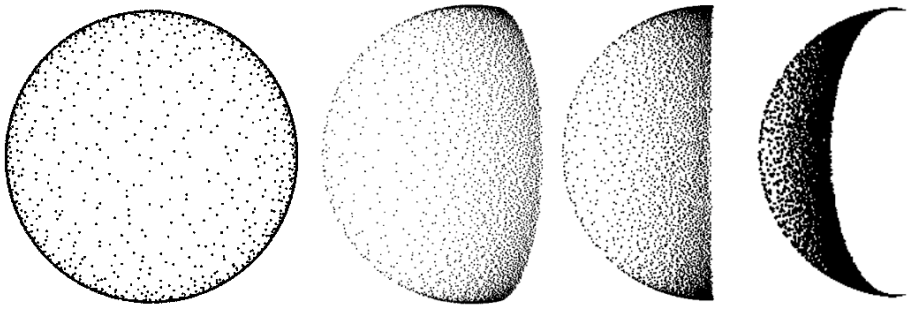

Fig. 11. Silhouette enhancement on a sphere, visualised from different angles.

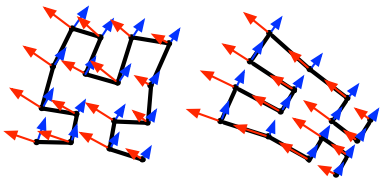

Fig. 12. Surface Hilbert curve. Left: aligned to parameterisation of geometry, which has an arbitrary relationship to the specified tensor field. Right: aligned to tensor field.

Control of the anisotropy can be useful to improve the accuracy of the approximation to the function by further increasing the efficiency of placement of a fixed number of samples. Anisotropic control allows samples to be placed closely in one direction, and farther apart in the orthogonal direction. This can be use-

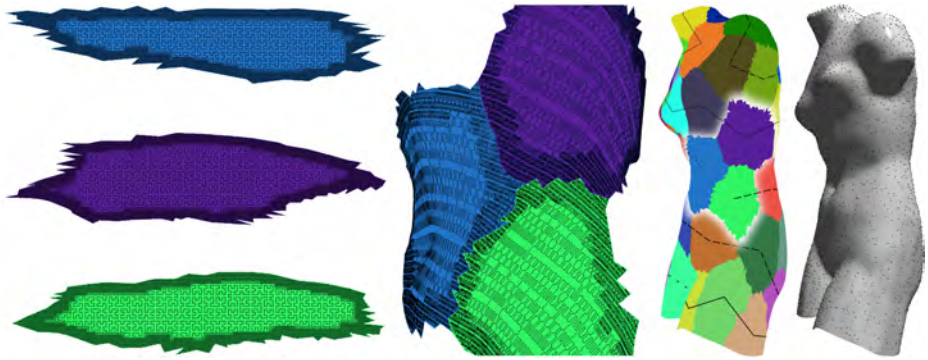

Fig. 13. Left to right: planar Hilbert curve segments, Hilbert curve segments mapped to the surface, Hilbert curve chains shown as different line styles, sampling.

ful, for example, when the principal curvatures of a manifold differ greatly, when sampling according to curvature, or for the placement of elongated elements, such as brush strokes.

Using a modified random walk method [32], M_A is segmented according to the tensor field. The aim of segmentation is to provide regions of the mesh which can locally be parameterised with a parametrisation whose orientation is in agreement with the local tensor field. Seed triangles are chosen on the mesh, and triangles are assigned to a seed based on the probability of a random walk from that seed arriving at that triangle. Probabilities of walking from one triangle to a neighbour are computed based on the similarity between the local tensor and the direction of walking; walking parallel to the primary eigenvector of the tensor is cheaper than walking orthogonally to it. Within each segment, singularities in the tensor field are then found per-triangle using a linear solver; the singularities are then connected by paths following the field. The mesh is then further segmented along these paths, as the tensor field cannot be aligned as desired in a segment containing a parametric singularity. Each segment is then parameterised anisotropically by minimisation of an energy function [33]. The solution is found using a sparse linear system, aligning and stretching the segment according to the local tensor field T. Each parameterised segment is then checked for alignment of parametrisation and tensor field with respect to an accuracy threshold, and to ensure that it does not self-intersect. If it fails to meet these requirements, it is further subdivided into smaller segments.

An individual isotropic Hilbert curve C is generated and clipped to each parameterised segment (see Fig 13). As the segment is already stretched with respect to the anisotropy, when this curve is mapped to the original mesh M_A, the anisotropy of T is induced onto C, resulting in an anisotropic set of curves on the mesh M_A. These curves are then sampled individually (see Fig 14).

The quality of the resulting sampling has been demonstrated in three ways. Firstly, several quantitative experiments were performed to ensure that neither the parameterisation nor the segmentation introduce artefacts into the output.

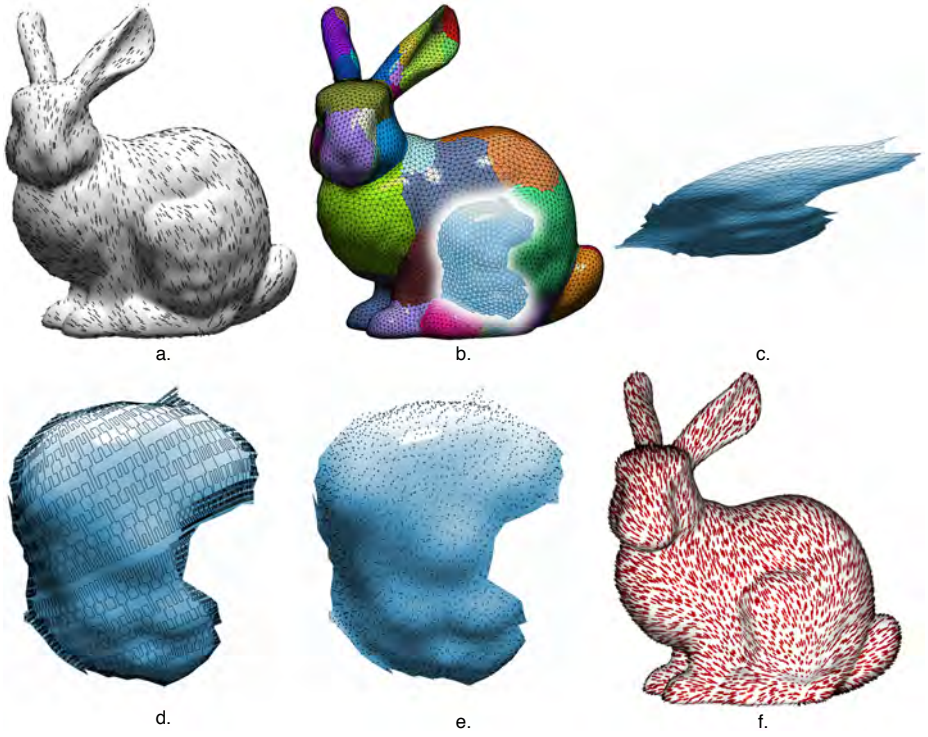

Fig. 14. Steps in our anisotropic sampling algorithm: (a) input tensor control field, (b) mesh segmentation, (c) parameterisation of one segment, (d) Hilbert curve on that segment (visualised at reduced depth), (e) point samples for that segment, (f) point samples as splats.

Secondly, anisotropic discrepancy experiments and a spectral analysis were performed (see Fig 15) using anisotropic sampling of a planar surface. They confirm that the sampling quality is consistent with that obtained in the isotropic case, and that the sampling has the expected anisotropy. When computing anisotropic discrepancy, the area of the sample rectangles is computed with respect to the metric defined by a constant tensor T in $[0,1]^2$. Thirdly, various tests were performed using differential domain analysis [23]. This method allows for spectral analysis of arbitrary mesh sample distributions, according to both a density function and anisotropy. Results show that our method correctly represents the input density functions and anisotropy, whilst maintaining the frequency response of stratified sampling methods. Fig 16 shows a differential domain analysis for isotropic mesh sampling; further results concerning anisotropic sampling are given in [19]. Fig 17 shows a splatting of the Dragon model, with and without anisotropy. Less white space is visible when it is anisotropically sampled as the elements better tessellate.

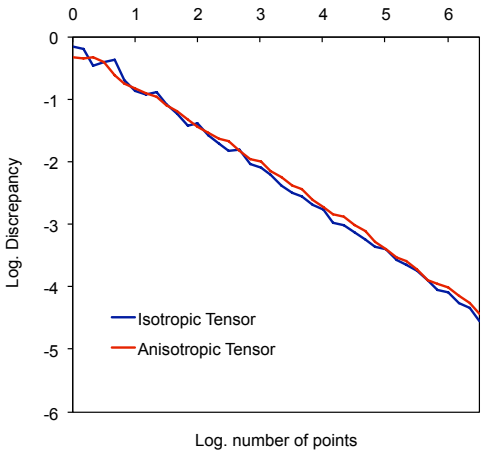

Fig. 15. Discrepancy of sample distributions generated with respect to an isotropic and anisotropic uniform tensor field.

5 Applications

In this section, we describe two applications of low discrepancy sampling, a stroke based rendering algorithm, and a demonstration of the potential use of sampling methods in shape retrieval.

5.1 Stroke Based Rendering

As one application of our methodology, we demonstrate a non-photorealistic rendering method that uses anisotropic samples generated by the algorithm described in Section 4.3. Various types of *brush* have been recorded by an artist, and manually scanned as graphics RGBA textures. *Brushes* include watercolours, crayons, pencils, inks, acrylics, pastels, and oils. For each brush type, 10 different stroke lengths are recorded, with 5 variations of each length. Samples are rendered as brush strokes in a fragment shader, the length of the stroke chosen according to the eigenvalue of the primary eigenvector in the tensor defined at the sample point. The stroke is then aligned to the primary eigenvector.

Fig. 18 shows a knot mesh, sampled with 30,000 points, rendered using watercolour paints, crayons, and pencils. Fig. 19 shows an Eight mesh, sampled with 20,000 points, rendered using inks, acrylic paints, and pastels. Fig. 20 shows a Venus mesh, sampled with 35,000 points, rendered using multiple ink tones, multiple pencil tones, and oil paints. Strokes are intentionally not clipped to the silhouette, in order to give a hand-created feel. The sampling method results in an equidistributed sampling, with local non-determinism, resulting in a well distributed set of strokes that look naturally placed.

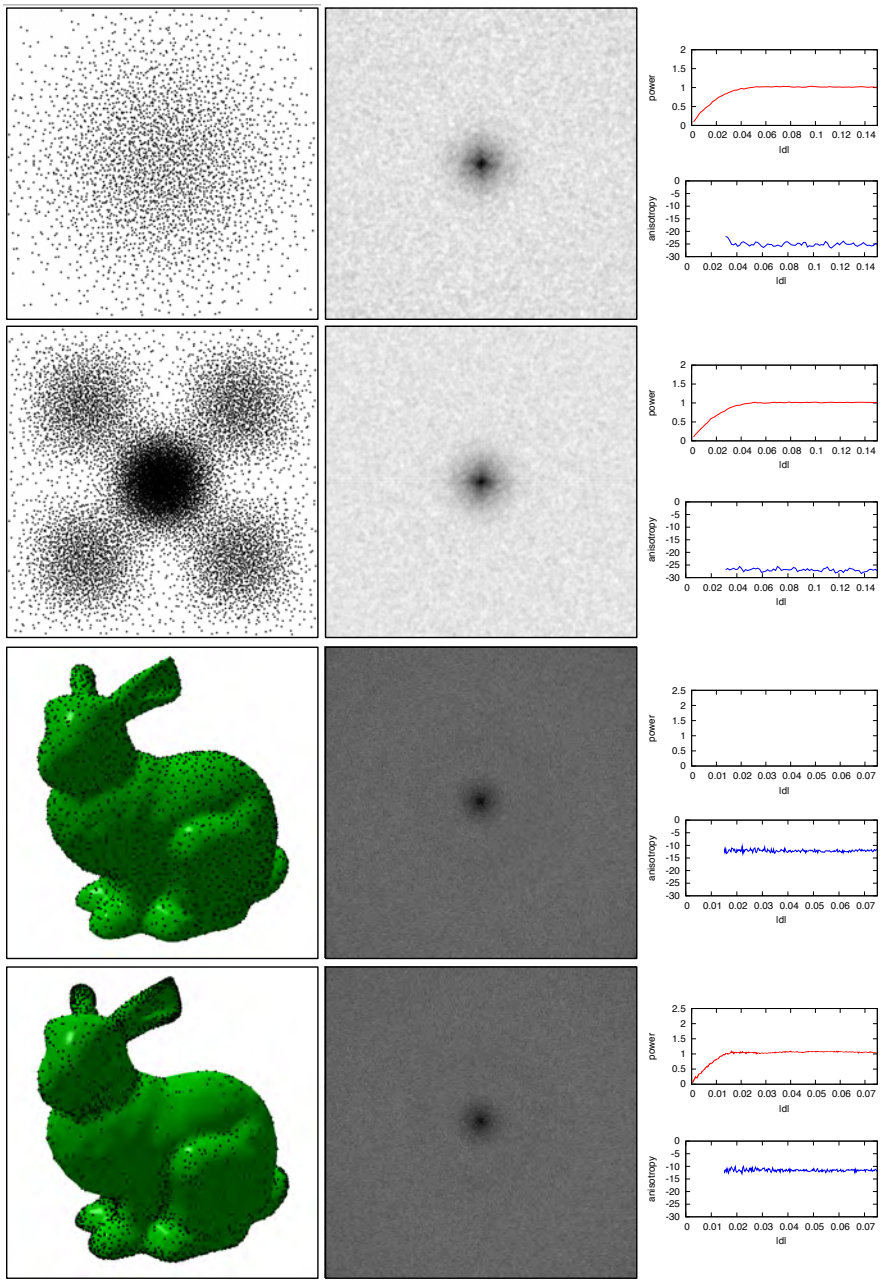

Fig. 16. Differential domain analysis. Isotropic, density-controlled sampling of 2D Gaussian blob (2,500 samples in $[0,2]^2$), Balzer function (10,000 samples in $[0,4]^2$), and uniform and mean curvature-controlled surface sampling of the Stanford Bunny (3,000 samples). Samples, spectrum, radial mean power, and anisotropy measure (constant value indicates an isotropic result). Results averaged over multiple runs.

Fig. 17. Dragon model. Left: splatted isotropically, right: splatted anisotropically.

Fig. 18. Knot mesh, 30k samples, rendered in left: watercolour, middle: crayon, right: pencil.

Fig. 19. Eight mesh, 20k samples, rendered in left: ink, middle: acrylic, right: pastel.

5.2 Shape Retrieval

Sampling methods are also of use in the problem of shape retrieval. Whilst any of the sampling algorithms could be used, we use the algorithm in Section 4.3 as it handles complex shape topologies better. A popular approach to shape

Fig. 20. Venus mesh, 35k samples, rendered in left: multi-toned ink, middle: multi-toned pencil, right: oil.

retrieval is to compute a set of feature descriptors for the shape, and then match these features to find similar shapes [34]. One particular approach is to compute distances between every pair of vertices in a mesh representing the shape, or some other set of sample points on the shape's surface, and to then make a histogram of those distances, which provides a summary of the shape [35]. Shapes can then be compared by computing a distance between their histograms.

Here, we do not attempt to provide a competitive algorithm for matching histograms, but, simply demonstrate the application of our sampling method to improve results over random sampling. We do this not directly in a shape retrieval application, but instead, simply match a model to a down-sampled version of itself, and investigate the error in terms of the distance between histograms. Obviously, shape matching can be performed more quickly if the same shape can be represented to within a given tolerance with fewer samples, or for the same number of samples, more robust matching can be performed.

For an input mesh, we compute the histogram of all pairwise distances between vertices; distances may be Euclidean or geodesic. We use a fixed number of 32 bins for all examples. The histogram of vertex distances is used as the *ground truth* for the mesh. We also sample the same mesh using our algorithm, and with a random sampling algorithm which takes into account variation in triangle areas. Samples generated for both methods are then snapped to the nearest vertex on the mesh. Otherwise, the corner-cutting nature of a linear ap-

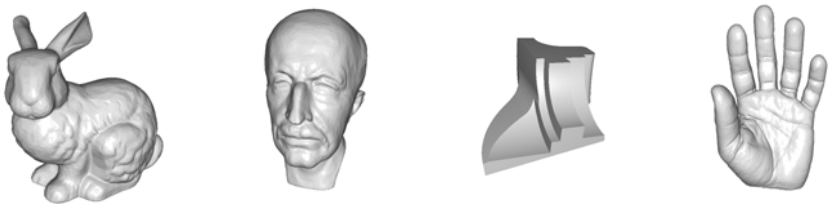

Fig. 21. Test meshes. Left to right: Stanford Bunny, Max Plank, Fandisk, Hand.

Table 1. Accuracy of histogram matching, for four test objects, comparing low-discrepancy sampling to random sampling. Relative errors as percentages, and maximum errors.

Histogram		**Bunny**		**Max Plank**		**Fandisk**		**Hand**	
		%	**max**	**%**	**max**	**%**	**max**	**%**	**max**
100 points	LD	1.997	0.0086	4.481	0.0159	1.463	0.0080	1.974	0.0073
	random	2.250	0.0090	5.485	0.0177	2.260	0.0117	2.543	0.0087
500 points	LD	1.068	0.0053	4.024	0.0133	0.629	0.0033	1.316	0.0042
	random	1.492	0.0063	4.390	0.0142	1.121	0.0036	1.436	0.0057

proximation to a surface as represented by a mesh would lead to a systematic error when comparing ground-truth distances measured between vertices. Also, distances between points which could lie in triangle interiors (distances would be underestimated in convex regions, and overestimated in concave regions).

A histogram is then constructed from the distances between samples, using the same bin widths as those used for the ground truth histogram. The ground truth and query histograms are then normalised to allow for differing numbers of points, and compared. We first compute the *relative error* between the histograms: $\sum |B_i| / \sum G_i$, where $|B_i|$ denotes the difference in heights of the i^{th} bins from the ground truth and sampled histograms, and G_i is the height of the i^{th} ground truth histogram bin. We also compute the *maximum error*, simply the maximum value of $|B_i|$ for any bin.

Fig. 21 shows the test models used: the Stanford bunny, Max Plank, Fandisk and Hand meshes. The Euclidean distance was used for most tests; geodesic distances gave broadly similar results. Figs 22–25 show how percentage and maximum error vary for the test models, for an increasing number of samples. For each sample size, 5 tests were run, and the average taken. Our sampling method converges to a baseline relative error faster and with less variability than random sampling, in all cases. The reduced variability means we can have more confidence that matches produced with low discrepancy samples are correct.

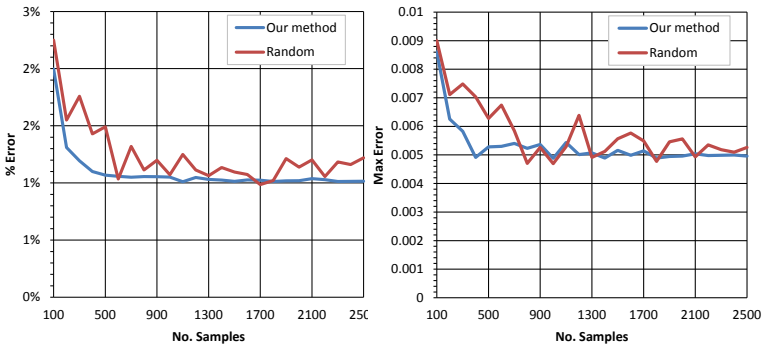

Fig. 22. Euclidean distance histogram matching for the Stanford Bunny mesh. Left: percentage error, right: maximum error.

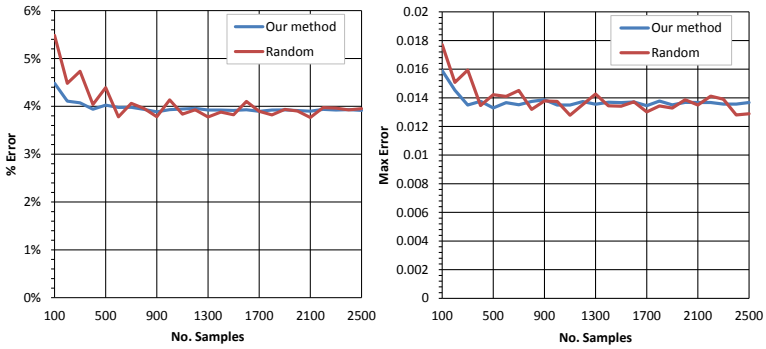

Fig. 23. Euclidean distance histogram matching for the Max Plank mesh. Left: percentage error, right: maximum error.

Both methods converge to a similar baseline relative error: the main advantage for low discrepancy sampling comes at lower numbers of samples. For maximum error, the behaviour is similar, with our method converging to the baseline error faster, and again with less variance than the random sampling method. Table 1 shows actual histograms for the four models, along with the percentage error and maximum errors for 100 and 500 point samples. In each case, the low discrepancy samples provide results closer to the ground truth.

Fig 26 shows relative and maximum errors for the Hand model, this time using geodesic distances, for an increasing number of samples. The geodesic metric is more useful for comparing models with many extremities, or complex topology. In this case, low discrepancy sampling again improves the results for relative and maximum errors, and indeed for the relative error, our method converges to a lower baseline error, although this does not occur for the maximum error.

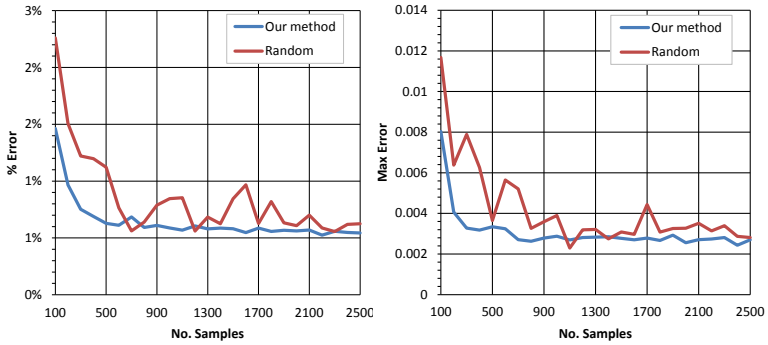

Fig. 24. Euclidean distance histogram matching for the Fandisk mesh. Left: percentage error, right: maximum error.

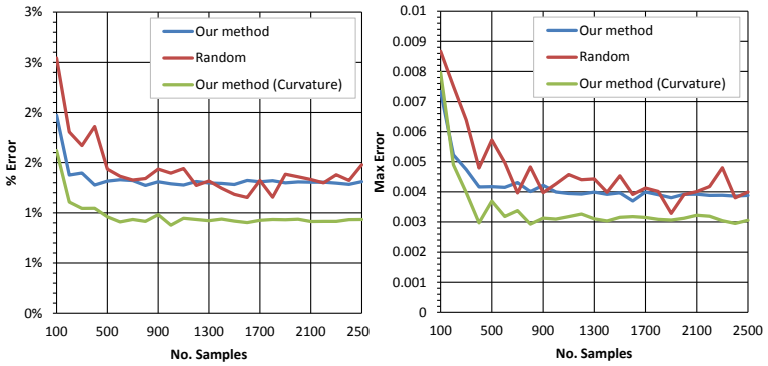

Fig. 25. Euclidean distance histogram matching for the Hand mesh. Left: percentage error, right: maximum error.

We believe the cause for convergence to a baseline non-zero error in the examples shown is due to the approach used to snap samples to the nearest vertex, which does not consider whether some other sample has already been snapped to the same vertex. In fact, if many more samples are used, both approaches do converge to zero error; for low discrepancy samples, this happens as the number of samples approaches the number of mesh vertices, whilst for random samples significantly more samples than mesh vertices are needed. Further work is needed to systematically analyse and eliminate this baseline error. One approach would be to leave sample points in triangle interiors rather than snapping them to a mesh point, but projecting them inwards or outwards by an amount depending on local surface curvature, to account for the mesh being a linear approximation to an underlying curved surface. While mesh vertices are assumed to lie on that surface, points inside mesh triangles generally do not, and constructing a histogram from them is certain to be biased if they are not corrected, as noted earlier

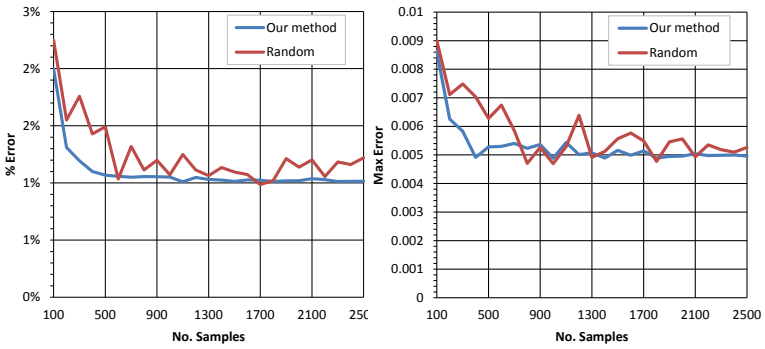

Fig. 26. Geodesic distance histogram matching for the hand mesh. Left: percentage error, right: maximum error.

It is also worth noting that the whilst the averaging of 5 test runs makes the random sampling results easier to interpret, it does hide the huge degree of variability in the results it produces—the low discrepancy sampling produces much more repeatable results with little variability. These results further confirm the property already noted that generalised stratified sampling provides a method of variance reduction (see Section 1), and demonstrate a practical improvement over random sampling.

6 Conclusions

This paper has summarised our earlier work on sampling parametric surfaces and meshes, the latter both isotropically and anisotropically, showing how low discrepancy samples can be generated by means of space-filling curves. These reduce the sampling problem from a difficult 2D problem to a much simpler 1D problem. We have also discussed two sample applications, rendering, and shape matching, showing how such low discrepancy sampling can potentially be useful in such problems.

References

1. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods. SIAM (1992)
2. Cook, R.L.: Stochastic sampling in computer graphics. ACM Trans. Graphics **5**(1) (1986) 51–72
3. Quinn, J.: Low-discrepancy Point Sampling of 2D Manifolds for Visual Computing. PhD thesis, Cardiff University (2009)
4. Floater, M.S.: Parametrization and smooth approximation of surface triangulations. Computer Aided Geometric Design **14**(4) (1997) 231–250
5. Hammersley, J.M.: Monte Carlo methods for solving multivariable problems. Ann. New York Acad. Sci. **86** (1960) 844–874

6. Halton, J.H.: On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numerical Math. **2**(2) (1960) 84–90
7. Cipra, B.: What's Happening in the Mathematical Sciences (1995–1996). Volume 3. Amer. Math. Soc., Providence, RI (1996)
8. Bowers, J., Wang, R., Wei, L.Y., Maletz, D.: Parallel poisson disk sampling with spectrum analysis on surfaces. ACM Trans. Graph. **29** (December 2010) 166:1–166:10
9. Cline, D., Jeschke, S., White, K., Razdan, A., Wonka, P.: Dart throwing on surfaces. Comp. Graph. Forum **28**(4) (2009) 1217–1226
10. Du, Q., Emelianenko, M.: Acceleration schemes for computing centroidal Voronoi tessellations. Numerical Linear Algebra with Applications **13**(2–3) (2006) 173–192
11. Liu, Y., Wang, W., Lévy, B., Sun, F., Yan, D.M., Lu, L., Yang, C.: On centroidal Voronoi tessellation – energy smoothness and fast computation. ACM Trans. Graphics **28** (2009) 101:1–101:17
12. Du, Q., Faber, V., Gunzburger, M.: Centroidal Voronoi tessellations: applications and algorithms. SIAM Review **41**(4) (1999) 637–676
13. Dobkin, D.P., Eppstein, D., Mitchell, D.P.: Computing the discrepancy with applications to supersampling patterns. ACM Trans. Graph. **15**(4) (1996) 354–376
14. Steigleder, M., McCool, M.: Generalized stratified sampling using the Hilbert curve. Journal Graph. Tools **8**(3) (2003) 41–47
15. Shirley, P.: Discrepancy as a quality measure for sample distributions. In: Eurographics. Elsevier Science, Amsterdam (1991) 183–94
16. Rovira, J., Wonka, P., Castro, F., Sbert, M.: Point sampling with uniformly distributed lines. Eurographics Symp. Point-Based Graphics (2005) 109–118
17. Hickernell, F.J.: A generalized discrepancy and quadrature error bound. Mathematics of Computation **67**(221) (1998) 299–322
18. Quinn, J.A., Langbein, F.C., Martin, R.R., Elber, G.: Density-controlled sampling of parametric surfaces using adaptive space-filling curves. In: Proc. Geometric Modeling and Processing. Number 4077 in LNCS, Springer (2006) 658–678
19. Quinn, J.A., Langbein, F.C., Lai, Y.K., Martin, R.R.: Generalized anisotropic stratified surface sampling. IEEE Transactions on Visualization and Computer Graphics **99**(PrePrints) (2012) 1
20. Nehab, D., Shilane, P.: Stratified point sampling of 3D models. In Gross, M., Pfister, H., Alexa, M., Rusinkiewicz, S., eds.: Symposium on Point-Based Graphics, Zürich, Switzerland, Eurographics Association (2004) 49–56
21. Ulichney, R.A.: Dithering with blue noise. Proceedings of IEEE **76**(1) (January 1988) 56–79
22. Bartlett, M.S.: An introduction to stochastic processes, with special reference to methods and applications. 2nd edn. Cambridge University Press (1966)
23. Wei, L.Y., Wang, R.: Differential domain analysis for non-uniform sampling. In: ACM SIGGRAPH 2011 papers. SIGGRAPH, New York, NY, USA, ACM (2011) 50:1–50:10
24. Li, H., Wei, L.Y., Sander, P.V., Fu, C.W.: Anisotropic blue noise sampling. ACM Trans. Graph. **29** (December 2010) 167:1–167:12
25. Sagan, H.: Space-Filling Curves. Springer-Verlag, New York, NY (1994)
26. Quinn, J.A., Langbein, F.C., Martin, R.R.: Low-discrepancy point sampling of meshes for rendering. In Botsch, M., Pajarola, R., Chen, B., Zwicker, M., eds.: Symposium on Point Based Graphics, Prague, Czech Republic, Eurographics Association (2007) 19–28
27. Saboret, L., Alliez, P., Lévy, B.: Planar parameterization of triangulated surface meshes. In: CGAL-3.2 User and Reference Manual. CGAL (2006)

28. Gu, X., Gortler, S.J., Hoppe, H.: Geometry images. ACM Trans. Graph **21**(3) (2002) 355–361
29. Meyer, M., Desbrun, M., Schröder, P., Barr, A.H.: Discrete differential geometry operators for triangulated 2-manifolds. In: VisMath III. (2002) 35–57
30. Cignoni, P., Rocchini, C., Scopigno, R.: Metro: Measuring error on simplified surfaces. Comp. Graph. Forum **17**(2) (1998) 167–174
31. Surazhsky, V., Gotsman, C.: Explicit surface remeshing. In: Symp. on Geom. Proc. (2003) 20–30
32. Lai, Y.K., Hu, S.M., Martin, R.R., Rosin, P.L.: Rapid and effective segmentation of 3D models using random walks. Computer Aided Geometric Design **26**(6) (2009) 665–679
33. Ray, N., Li, W.C., Lévy, B., Sheffer, A., Alliez, P.: Periodic global parameterization. ACM Trans. Graph. **25**(4) (October 2006) 1460–1485
34. Tangelder, J., Veltkamp, R.: A survey of content based 3d shape retrieval methods. In: Shape Modeling Applications, 2004. Proceedings. (2004) 145–156
35. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Shape distributions. ACM Trans. Graph. **21**(4) (October 2002) 807–832