The Visual Computer manuscript No.
(will be inserted by the editor)

Streaming Surface Sampling using Gaussian ϵ-nets

Pablo Diaz-Gutierrez · Jonas Bösch · Renato Pajarola · M. Gopi

Received: date / Accepted: date

Abstract We propose a robust, feature preserving and user-steerable mesh sampling algorithm, based on the one-to-many mapping of a regular sampling of the Gaussian sphere onto a given manifold surface. Most of the operations are local and no global information is maintained. For this reason, our algorithm is amenable to a parallel or streaming implementation, and is most suitable in situations when it is not possible to hold all the input data in memory at the same time. Using ϵ-nets, we analyze the sampling method and propose solutions to avoid shortcomings inherent to all localized sampling methods. Further, as a byproduct of our sampling algorithm, a shape approximation is produced. Finally, we demonstrate a streaming implementation that handles large meshes with a small memory footprint.

Keywords normal quantization, surface sampling, shape approximation, epsilon-nets

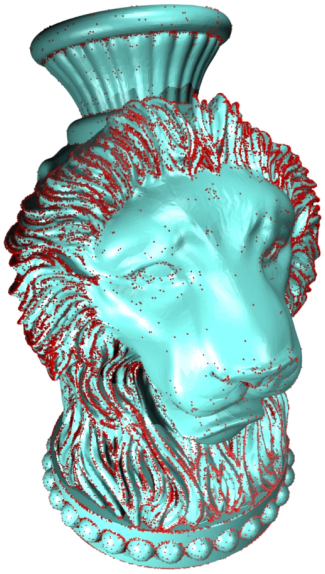

Fig. 1 Feature sensitive samples on a mesh.

1 Introduction

Polygon mesh sampling is important in many geometry processing problems, including shape approximation, surface reconstruction and parameterization. Correctly sampling surfaces involves choosing a set points such that their interpolation faithfully reproduces the desired features of the given surface, both in terms of geometry and topology. In this paper, we propose an efficient algorithm to produce a feature sensitive sampling of a surface. The output sample set is a subset of the vertices from the input surface.

The fundamental challenge in surface sampling is to find the minimal set of sample points that capture the features of a shape within an error bound. For shape approximation, Clarkson [2] proved that the minimum sampling size is proportional to the integral of the absolute Gaussian curvature over the surface. Our method computes a regular sampling of the Gaussian sphere, and then selects as samples all the points from the input surface whose surface normal coincides with one of those Gaussian sphere samples (a one-to-many mapping

Pablo Diaz-Gutierrez, M. Gopi
University of California, Irvine
6210 Donald Bren Hall
Irvine, CA 92697-3425
E-mail: {pablo,gopi}@ics.uci.edu

Jonas Bösch, Renato Pajarola
University of Zürich
Department of Informatics
Binzmhlestr. 14
8050 Zürich, Switzerland
E-mail: boesch@ifi.uzh.ch, pajarola@acm.org

from the Gaussian sphere to the given surface). This method can be justified using the theory of ϵ-nets, a construction that guarantees sufficient and well placed surface samples given a distance function. The result is a simple, robust surface sampling algorithm with a small memory footprint. This algorithm has potential uses in efficient shape approximation, parameterization and compression. Finally, we describe an application of our sampling algorithm as a fast shape approximation technique.

1.1 Main Contributions

Following are the main contributions of this paper:

- We introduce the sampling of a surface as the one-to-many mapping of a sampling on Gaussian space onto the surface, and justify it using the theory of ϵ-nets.
- We present a practical algorithm that applies the above theory for sampling polygonal meshes, while allowing the user to interactively control the placement and density of the samples.
- We provide a detailed analysis of our sampling algorithm and its properties, among which we highlight its ability to handle large meshes with a small memory footprint.
- We use the vertex elimination stage of our sampling method as a fast algorithm for the problem of shape approximation.

After a brief survey of previous work in Section 2, we discuss the theory and our sampling algorithm for smooth manifolds in Section 3. We adapt this theory to manifold meshes in Section 4 and discuss the application of this method to the problem of shape approximation. Finally, in Section 5 we describe the implementation details of our sampling method in a streaming framework.

2 Previous Work

In this section we briefly survey the extensive literature on relation with to our proposed surface sampling method and its relationship with the problem of shape approximation.

Surface Sampling: Sampling for surface reconstruction has been widely studied in the literature [4]. Although different in nature, surface reconstruction shares with surface sampling the need for a solid theoretical foundation regarding error measures. However, the similarities cannot go very far, since the type of data and constraints present in surface reconstruction and shape simplification are fundamentally different. With the exception of some local methods like [1,7], most surface reconstruction algorithms are based on global constructions like the medial axis of the object and its relationship to 3D Voronoi diagrams. Since these algorithms are inherently global, they pose significant resistance to their parallel or streaming implementation.

Gaussian sampling algorithms are those which use normal vectors for geometry processing. As illustrated in [18], Gaussian sampling also provides a means to efficiently select data points with *interesting* features without costly distance computations. One of the most singular advantages of Gaussian sampling is its amenability for online and streaming algorithms. We exploit this feature in the streaming implementation of our sampling algorithm. Other applications of Gaussian maps include curvature based mesh segmentation [20, 3] which, as we continue reviewing below, provides a theoretically sound foundation for shape approximation methods.

Sampling for Shape Approximation: Discrete shape approximation techniques produce a sampling of the most important points on the surface. These points can be a subset of those in the input, or they can be relocated to an optimal position at greater computing cost. Our sampling method falls in the former category. Because our approximation algorithm is a byproduct of our surface sampling method, we consider only samples that are on the input surface. Since there is a long history of surface simplification algorithms, we refer to excellent surveys in this field [9,12]. In general, these methods try to optimize an energy functional or iterate in order to find the (optimal) positions and shape of the mesh elements (vertices, edges and faces) that would reduce the *approximation error* [3,17]. The main difference between most of these methods and ours is that they require access to a relatively large amount of connected geometry before deciding when and how to simplify. On the other hand, we decide whether to keep or eliminate a vertex solely based on local information.

Finally, we must refer to the fundamental theoretical basis for our algorithm. Recently developed by Clarkson [2] using results by others [8,13,16], the theory of ϵ-nets represents a solid foundation for the joint study of surface sampling and triangulation algorithms.

3 Sampling of Smooth Manifolds

The following are desirable properties for a sampling S of a smooth surface U, using a chosen metric function:

1. **Anisotropy**: At any point on the surface, sampling should be denser along the higher curvature direc-

tion than along the lower curvature direction. This is most naturally accomplished by using a distance function that grows faster along higher curvature directions. Under this configuration, samples would be at a uniform ϵ distance from each other, when measured by the appropriate metric.

2. **Sufficiency**: Every surface point $x \in$ U should not be farther than an ϵ distance from the closest sample $p \in$ S, as measured by the distance function chosen for the previous condition. This condition determines the correctness of the sampling, ensuring that all the features of the sampled surface are properly captured.
3. **Minimality**: Every sample $s \in$ S should be necessary, in the sense that removing one of them could break the condition of sufficiency. In practice, this condition plays the role of preventing unnecessarily dense sampling, making sure that samples are not closer than an ϵ distance from each other.

In this section we first briefly describe the theory of ϵ-nets as developed by Clarkson [2], followed by our choice of distance functions and our sampling algorithm.

3.1 ϵ-net Theory

Let U be a surface and $D(x, y)$ be the distance metric on U that measures the distance between two points x and y on U. Some sampling properties, which lead to the concept of ϵ-nets, are defined below.

Definition 1 **ϵ-cover:** Given a surface U and a set of samples S $\in$ U, if for every point $x \in$ U there exists a sample point $s \in$ S within a distance ϵ ($D(x, s) \leq \epsilon$), then the sampling S is called an ϵ-cover of surface U.

An ϵ-cover S of surface U meets the condition of *sufficiency* by definition. Moreover, a sample set T of surface U is an ϵ-cover of U if it contains a sample set S which is also an ϵ-cover of U. Although finding the smallest of all possible ϵ-covers is NP-hard, in order to meet the condition of *minimality* we can start with a sufficient sampling and simply remove all the unnecessary samples.

Definition 2 **ϵ-packing:** Let S be a sample set on surface U, and $s, t \in$ S be any two samples. If $D(s, t) \geq \epsilon$, then S is called an ϵ-packing of surface U.

Definition 3 **ϵ-net**: If sample set S is both an ϵ-cover and an ϵ-packing of U, then S is an ϵ-net of U.

3.2 Choice of Distance Function

As previously studied [2], although the fundamental theory behind ϵ-nets is independent of the choice of metric for their construction, different distance functions confer the structure with desirable properties for various applications. We have studied three possibilities for our distance function, with their qualities and shortcomings illustrated in Figure 2. In the context of surface approximation, the Euclidean metric $\mathcal{L}_2$ generally produces a low Hausdorff distance between the input surface and its approximation. However, it misses important features where the curvature changes rapidly. The $\mathcal{L}_{2,1}$ norm measures the distance between two surface points as the Euclidean distance between the surface normal vectors at these points: $D(x, y) = ||n(x) - n(y)||$. This distance function appropriately captures the curvature variations and features, but it produces a more significant deviation from the original surface in low curvature regions. Finally, the isophotic metric, which is a convex combination of the L_2 and $L_{2,1}$ distances is chosen for its adaptability in [2,3]. This distance function provides a good balance between the two previously mentioned alternatives. Unfortunately, the construction of ϵ-nets using either an isophotic or Euclidean metric requires access to a neighborhood within a radius of at least ϵ around each sample. Having stated our goal of producing a localized sampling and approximation method, we choose the $\mathcal{L}_{2,1}$ norm, which can be computed by looking only at the immediate neighborhood around each sample.

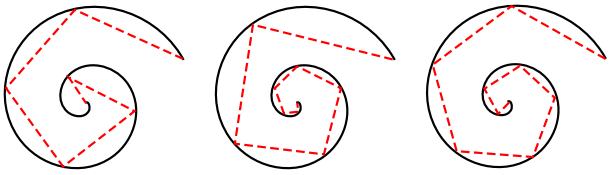

Fig. 2 Schematic illustration of three metrics considered for ϵ-net construction. The black spirals represent the surface being sampled, and the corners of the red dashed polylines indicate the position of the samples. The metrics being used are, from left to right: Euclidean, $\mathcal{L}_{2,1}$ and isophotic.

It has been shown [8,2] that the size of an ϵ-net of surface U using the $\mathcal{L}_{2,1}$ metric is proportional to $\int_{\mathrm{U}} |K(x)| dx/\epsilon^2$, where $K(x)$ is the Gaussian curvature at the point x. That is, *the sampling size is proportional to the total absolute Gaussian curvature of the surface*. Given these properties of anisotropic sampling, amenability for local decision making and control over the sampling density, we choose $\mathcal{L}_{2,1}$ to be our distance function, and this constitutes the basis for our sampling algorithm.

3.3 Conceptual Sampling Algorithm

Since $\mathcal{L}_{2,1}(x, y) = ||\mathbf{n}(x) - \mathbf{n}(y)||$, the ϵ-net sampling of $\mathcal{L}_{2,1}$ on a convex surface U is equivalent to an Euclidean distance ϵ-net on the Gaussian sphere. In order to exploit this duality, we construct a uniform tessellation of a unit sphere by iterated regular subdivision of the faces of a regular polyhedron. Let l be the distance between any two adjacent vertices of this tessellation. The vertices are at least $l - \mu$ away from each other, for a small positive μ, and any point on the sphere is not farther than $l - \mu$ distance away from the closest vertex in the tessellation. In other words, the vertices of this tessellation form an Euclidean metric $(l-\mu)$-net sampling of a sphere.

Given the vertices of a uniform tessellation of a unit (Gaussian) sphere, *our sampling of a surface U is given by the sample set S, which consists of all those points $x \in U$ whose surface normal $n(x)$ coincides with any of the vertices of the Gaussian sphere tessellation.* Different orientations of the Gaussian sphere produce different sample sets on surface U. However, all these sample sets are equivalent, in the sense that they satisfy the same sampling properties and error bounds. These sample sets are always an ϵ-cover of surface U using the $\mathcal{L}_{2,1}$ metric, indicating that the samples meet the condition of sufficiency. Further, if there is enough separation among samples, then the sample set is also an ϵ-packing, and thus an ϵ-net, meeting the condition of minimality. The condition of minimality is automatically satisfied for a special class of shapes, defined in Section 3.4. For other shapes, we remove superfluous vertices while still maintaining an ϵ-cover.

3.4 Sampling Properties

The size of the sampling produced by the above method is proportional to the total absolute Gaussian curvature. For convex genus-zero objects, the Gaussian curvature is positive everywhere, and the total absolute Gaussian curvature is the surface area of the unit sphere, 4π (given by the Gauss-Bonnet theorem [14]). Using the following definition of convexity, this result can be extended to two-manifolds with higher genus, considering objects like a torus to be convex for our purposes.

Definition 4 A compact two-manifold M_2 in R^3, with genus g, is 2-convex *iff* its total absolute Gaussian curvature is $4(g + 1)\pi$.

Lemma 1 *A convex two-manifold M is also 2-convex. The converse is not necessarily true.*

By Definition 4 and Lemma 1, the manifolds in Figure 3 are convex and hence 2-convex. The tori in Figure 6 are 2-convex, but they are not convex. Correspondingly, the dimpled sphere in the same figure is neither convex nor 2-convex.

Since the total absolute Gaussian curvature is the same for any 2-convex model with a given genus g, regardless of any other geometric considerations, the total number of samples should also be the same for all these models (for a given tessellation of the Gaussian sphere). Also, for a given value of ϵ (which defines the tessellation level of the Gaussian sphere), the sampling produced by our method is minimal for this class of models. The reason is that, for any value of the surface normal n, on a smooth, 2-convex surface there can be exactly $g+1$ surface points with surface normal n. Consequently, there are exactly $g + 1$ pre-images for every Gaussian vertex. Removing any one of these pre-image surface points from the sample set would produce a surface region which is farther than ϵ from the closest sample, this way rendering the sample set not sufficient.

Figures 3 and 4 illustrate this property. In the former, we show a sequence of models depicting the transformation of a sphere into a rounded cube. In the latter, the sampled surface is static, but the Gaussian sphere is smoothly rotating. As expected, in both cases the number of samples remains constant after each transformation. Moreover, the distribution of these samples exhibits anisotropy, with more samples migrating towards the high curvature regions. The maximum normal deviation between the adjacent samples is constant by construction. Figures 4 and 5 also illustrate a degree of user control over the sampling process. The sampling density depends on the refinement of the Gaussian sphere tessellation (Figure 5), while the placement can be partially controlled through appropriate rotations of the Gaussian sphere (Figure 4).

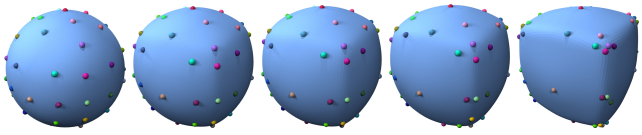

Fig. 3 As a sphere morphs into a rounded cube, the samples from an ϵ-net using $\mathcal{L}_{2,1}$ metric migrate towards high curvature regions. Surface samples are colored based on their associated Gaussian sample, with consistent coloring throughout the sequence. Since all the shown surfaces are convex, the total number of samples is constant.

The properties of the application of our sampling method to higher genus objects is illustrated in Figure 6, which shows the samples obtained on two (convex) tori of different sizes. In both cases, the number

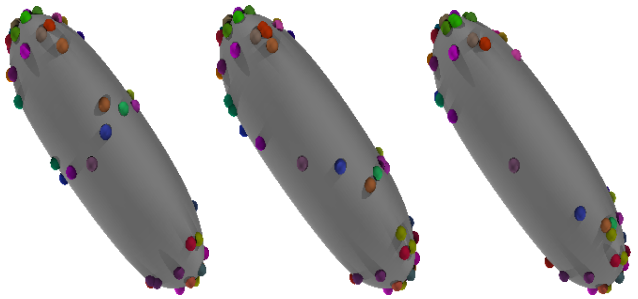

Fig. 4 Samples on a shape, produced with varying orientations of a Gaussian sphere tessellation. The surface samples smoothly slide over the surface following the rotation of the Gaussian samples. Since the surface is convex, the number of samples is constant. Note also the effect of the anisotropic $\mathcal{L}_{2,1}$ metric on the sample placement.

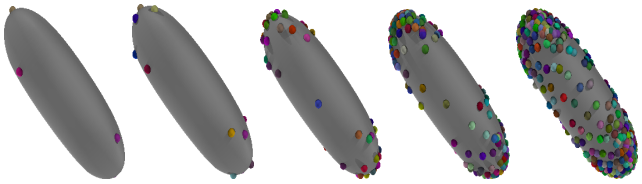

Fig. 5 Samples on a shape, produced with increasingly refined tessellations of the Gaussian sphere. The sequence of Gaussian sphere tessellations starts with an octahedron and continues by splitting each triangle in four, at each iteration. The produced samples accumulate faster on high curvature regions.

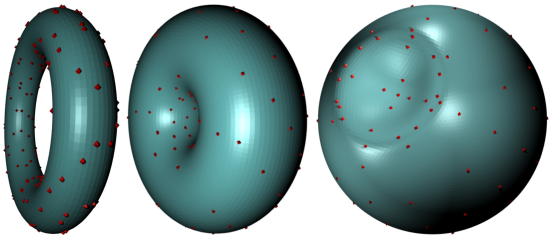

Fig. 6 Samples produced on different manifolds, computed as the surface points whose normal coincides with that of a Gaussian sample in the tessellation of a sphere. The sampling size produced by our algorithm is proportional to the total absolute Gaussian curvature. For any 2-convex manifold (left, middle), this value is constant and proportional to its genus. Concavities in a shape (right) increase its total absolute Gaussian curvature, and hence the model would require more samples than if it were convex.

of samples is the same, but the different distribution of the Gaussian curvature in the thicker torus makes the samples migrate towards its interior. Further, as a consequence of Definition 4, for a given tessellation of the Gaussian sphere, the total number of samples on either torus is twice the number of samples on a convex genus zero object. On the other hand, the dented sphere in Figure 6 illustrates the effect of concavities in the sampling size. The total absolute Gaussian curvature of the model shown is higher than that of the convex objects from Figure 3, and therefore more samples are produced to maintain the proportionality.

3.5 Non-convex sampling

Our sampling method is based on a quantization of the surface normals. In fact, the sampling size is proportional to the absolute Gaussian curvature as measured using not the original normal vectors, but the quantized normal vectors. When the Gaussian sphere tessellation is refined, the quantized normal vector approaches the exact normal vector, and hence the Gaussian curvature measure also approaches the correct value. In many common situations, there may be differences. If a genus g manifold M is *2-convex* by Definition 4, the mapping of its normals onto the Gaussian sphere wraps around the sphere exactly $g+1$ times. But, if M is not 2-convex, then its total absolute Gaussian curvature is higher than the $4(g+1)\pi$, and this mapping contains *folds*. The resolution and orientation of the Gaussian sphere tessellation determines whether these folds are captured in the normal quantization and are able to contribute samples. Accordingly, sampling a surface which is not 2-convex may or may not satisfy the ϵ-packing condition, and hence the condition of minimality. Nevertheless, even in the case when the quantized estimate of the total absolute Gaussian curvature differs from the real value, the resulting sampling is still an ϵ-cover, and hence it meets the condition of sufficiency. In Section 4.2 we will discuss how to achieve an ϵ-packing in such models, and therfore an ϵ-net.

4 Sampling Manifold Meshes

In this section, we present an algorithm that applies the introduced theory to the problem of sampling polygonal meshes, including those with sharp features. We first suggest a direct numerical approach and then a more robust approach, based on the quantization of polygon normals.

4.1 Direct Mesh Sampling

On a *smooth* manifold M, its samples are all the points on M whose normals coincide with one of the vertices of the Gaussian sphere tessellation G. On a polygonal approximation of M, the samples can ideally be on the mesh faces, edges, or vertices. Since each mesh face represents one normal on the Gaussian sphere, and each mesh edge represents a "curve" of normals, the likelihood of these normals coinciding exactly with a Gaussian vertex is essentially zero. Thus the samples can come only from the input mesh vertex set. A mesh vertex v represents the range of normals N_v that span

the interior of a spherical polygon defined by the normals of the mesh faces incident on v. Mesh vertex v is considered a sample if and only if its induced spherical polygon on the Gaussian sphere G contains one or more vertices of G (see Figure 7).

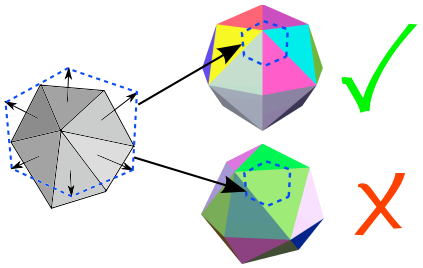

Fig. 7 Direct mesh sampling. A mesh vertex is a sample if the spherical polygon formed by its incident face normals contains a Gaussian vertex.

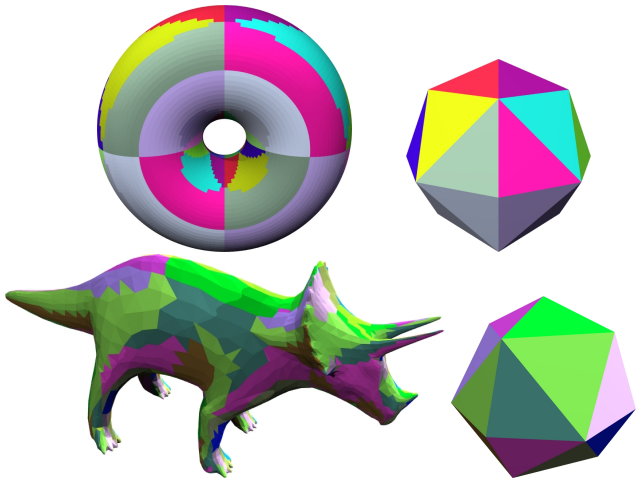

Fig. 8 The faces of two meshes (left) are clustered according to their associated Gaussian triangles (Gaussian sphere shown to the right). These clusters are separated by *feature edges*. Vertices adjacent to any *feature edges* are *candidate samples*.

Unfortunately the spherical polygons created by connecting the normal vectors of the incident triangles are small (in low curvature regions) and may self-intersect (in saddle vertices). Point location in such spherical polygons is numerically unstable and best avoided. Instead, we proceed as described below.

4.2 Conservative Mesh Sampling

A more robust sampling method starts considering a wide range of mesh vertices as candidate samples, and then discards those that can be positively shown not to map to a sample in the Gaussian sphere tessellation.

Definition 5 Gaussian triangle association, feature edge and candidate sample: A mesh face t with normal n_t is *associated* with a Gaussian triangle t_G if n_t is located inside t_G in normal space. A mesh edge e is a *feature edge* if its two incident faces are associated with different Gaussian triangles. A mesh vertex v is a *candidate sample* if it is incident on any *feature edge*.

Given a specific tessellation G of the Gaussian sphere, let us assign each mesh face to its associated Gaussian triangle. This would partition the input mesh into regions with the same associated Gaussian triangle. Under this partitioning, it can be seen that the surface samples are a subset of all the mesh vertices on the boundaries between partitions. We prune the candidate sample set by applying a simple filtering rule as follows.

Identifying non-samples: Not all candidate samples are samples. For example, let a, b, and c be Gaussian triangles such that a is edge-adjacent to b and b is edge-adjacent to c. Let A_1, B_1, C_1, B_2, A_2 be mesh faces incident on a mesh vertex v, in order. Let A_1 and A_2 be associated with a, B_1 and B_2 be associated with b, and C_1 be with c, via their normals. There are feature edges between A_1B_1, B_1C_1, C_1B_2 and B_2A_2, all incident on v. Hence v will be considered a candidate sample. Clearly, the spherical polygon formed by the normal vectors of the mesh faces around v does not enclose any Gaussian vertex and hence v cannot be a sample. Such cases can be generalized as follows (see Figure 9). Consider the spherical polygon formed by the normal vectors of the faces incident on the mesh vertex v. If consecutive vertices of this spherical polygon fall in adjacent [1] Gaussian triangles, and if this spherical polygon can be closed without enclosing a Gaussian sample,then v cannot be a sample. We call these vertices, as well as all the mesh vertices that are inside the partitions, "non-samples". The rest of the mesh vertices are the samples chosen by our algorithm.

From ϵ-cover to ϵ-net: The samples that remain after the above operation clearly form an ϵ-cover of the input model with respect to the $\mathcal{L}_{2,1}$ metric, since the normal deviation between any point on the original model and its closest sample is less than ϵ (the angle between adjacent Gaussian vertices). But if the input model is not 2-convex, under the various situations explained in Section 3.4, the normal deviation between two adjacent samples on the mesh may also be less than ϵ. In this case, the sampling may not be an ϵ-packing (and hence not an ϵ-net). Often, slightly perturbing the orientation of the sphere mitigates this issue. But this

[1] The condition of adjacency through edge connectivity can be relaxed to ensuring that the union of Gaussian triangles traversed by one edge of the spherical polygon form a convex polygon.

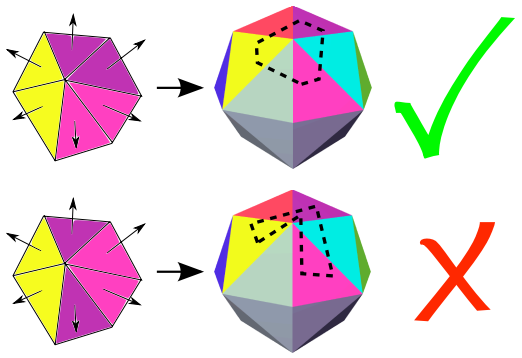

Fig. 9 Candidate samples (left) can be pruned if the normals of their incident faces form a spherical polygon (middle) that can be shown not to contain a Gaussian sample just by looking at their associated Gaussian triangles.

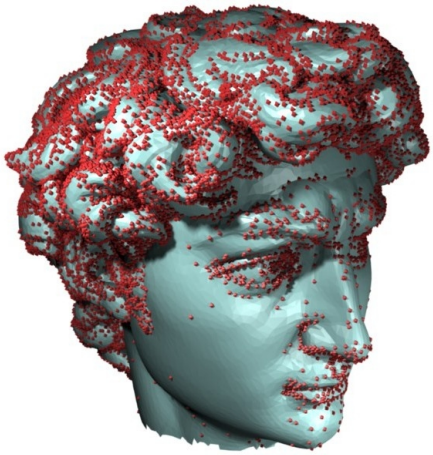

Fig. 10 Feature sensitive samples produced by our algorithm on a large mesh. The Gaussian sphere was tessellated using 112 triangles.

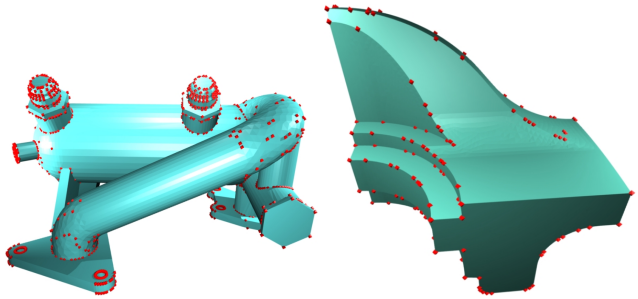

Fig. 11 Samples produced on two mechanical parts. Areas with null Gaussian curvature (cylindrical and planar regions) do not contain any samples.

procedure is not robust and it does not always guarantee an ϵ-packing in all regions of the input surface. Instead, we propose an optional iterative process that can be used to eliminate unnecessary samples and turn such a ϵ-cover into an ϵ-net.

This optional process requires that every non-sample vertex is merged with an arbitrary adjacent vertex [10] (with constant time per vertex), thus maintaining the triangulation of the input mesh with only the retained samples. On this modified mesh, if we rerun the sampling algorithm using a different orientation of the Gaussian sphere, then we can eliminate some samples that violate the ϵ-packing condition. We can repeat the process until convergence. The resulting sample set is both an ϵ-cover and also an ϵ-packing, and hence an ϵ-net of the original mesh.

All the operations performed by our sampling algorithm, namely identifying and removing non-samples and the optional edge collapse operations, are local operations around the mesh vertices. Hence the entire sampling algorithm requires no communication/sharing of data between different operations. Further, each individual operation requires only local memory access to the vertices adjacent to a candidate sample. Our algorithm is thus embarrassingly parallelizable. In the following section, we describe a streaming version of our algorithm that exploits these properties.

5 Streaming Implementation

The streaming implementation of the above Gaussian sampling and approximation algorithms is based on an underlying *streaming meshes* representation [11] for polygonal surfaces. Similar to the stream-processing approach of [15], the (streaming) triangle mesh is processed sequentially from out-of-core with only a limited amount of data kept active in main memory at any time. As indicated in Figure 12, within a sliding window over the streaming mesh data the active vertices and triangles are processed in a multi-stage pipeline of sampling and simplification operators.

The three major phases of the stream-processing pipeline are the identification of sample and non-sample vertices as outlined in the previous section, the removal of non-sample vertices from the input mesh, and the reestablishment of the proper (streaming) mesh output format.

In the first stage, the streaming mesh input is converted into a half-edge triangle mesh data structure [19] for efficient mesh manipulation in main memory. Care has to be taken to correctly maintain and treat references to future or past mesh elements – in the streaming order – while processing the in-core triangles. Hence the active region of the streaming mesh (see Figure 12) is maintained in-core in a proper topological mesh data structure that dynamically changes as the stream border advances. Moreover, after mesh initialization, each vertex is processed to check if no feature edge is incident on it (inside the partition), or if feature edges are incident on it but the vertex is a non-sample (see

Section 4). These operations are local and can thus be applied in a streaming context.

In the second processing stage, non-sample vertices are pruned from the input mesh by the application of mesh simplification operations. Non-sample vertices are removed by an iterative and greedy application of half-edge collapses [5]. In order to get a better approximation of the model, we retain a sufficient number of non-samples in-core and repeatedly choose the best half-edge to collapse from all the half-edges that are in-core. Such an approach constitutes an improvement over a greedy method, since each active non-sample vertex gives rise to a number of half-edge collapse candidates. Only half-edge collapses which fulfill a set of mesh topology as well as normal deviation constraints can become collapse candidates. Dynamically maintaining a priority queue of applicable half-edge collapse candidates, prioritized using a quadric error metric [6], the collapse which introduces the smallest error can be selected efficiently.

During the streaming process, triangles in every stage are kept in a priority queue so that faces passed to the next stage are always monotonically ascending according to their minimal vertex corner coordinates along the streaming-axis. When a half-edge is collapsed, its mesh elements (its incident faces and their half-edges) are removed from the mesh immediately. Also, all modified faces (those incident on a removed vertex) are reinserted into the priority queue.

Finally, the last stage is formed by a data cleansing process, in which faces are re-indexed to account for the eliminated non-sample vertices, and non-referenced vertices are omitted from the output.

During the entire process of sampling, edge-collapse, and streaming mesh I/O, the necessary conditions on triangle mesh neighborhood existence are maintained, such as minimum and maximum extents along the streaming-axis for the triangle itself and its one- and two-ring neighborhoods. The output is a simplified mesh, in the same streaming mesh format as the input, where all the vertices are samples. This output can be fed back to the system to iterate this method over a different, random orientation of the Gaussian sphere to ensure the sampling is an ϵ-net on the original mesh.

6 Results

As indicated in the introduction and our elaborations on the proposed sampling of manifold surfaces, the main contribution of this paper is a local, feature sensitive sampling of manifold surfaces. Our implementations demonstrate the effectiveness of the proposed sampling method on polygonal, i.e. triangle meshes.

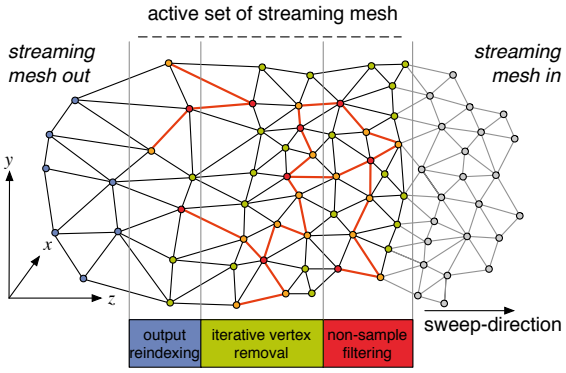

Fig. 12 Streaming pipeline with Gaussian sample filtering and non-sample vertex removal.

The adaptivity of our sampling method to sharp features or high curvature regions of a surface is demonstrated throughout the paper, and in particular is shown in Figures 1, 10, 13 and 14. From a qualitative aspect we can thus observe that the experimental sampling results perfectly match the given theory as samples are concentrated where indicated by the Gaussian sphere sampling and surface curvature.

Fig. 13 A mesh of 345,944 faces sampled using 30,392 feature sensitive samples.

From a quantitative point of view it is assured, by construction of the sampling algorithm in Section 4, that we identify a set of samples on the surface that form an ϵ-cover with respect to the $\mathcal{L}_{2,1}$ metric.

Through the application of our sampling algorithm to shape approximation we can further evaluate it quantitatively to a mesh simplification baseline, i.e. QSlim. While the proposed sampling addresses the $\mathcal{L}_{2,1}$ normal error metric, the typical comparison for mesh simplification is based on the $\mathcal{L}_2$ norm Euclidean distance metric between the original and simplified surface. We

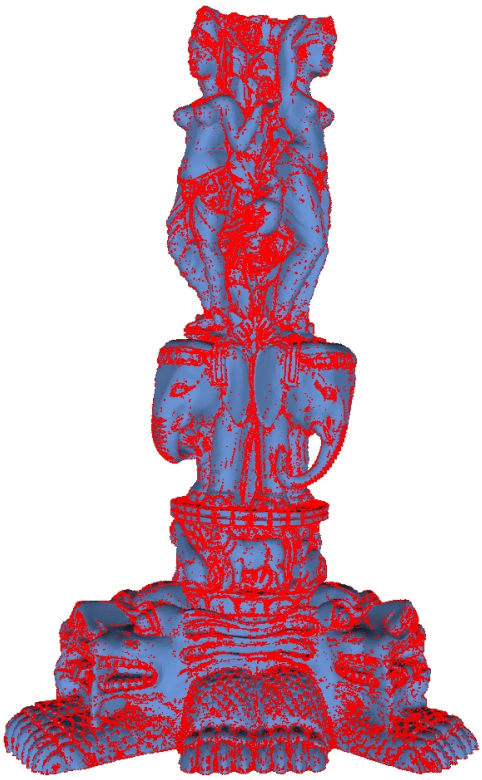

Fig. 14 A mesh of 10,000,000 faces sampled using 883,505 feature sensitive samples.

Model	Faces		Error	
	original	simplified	ours	qslim
Armadillo	346K	200K	0.0012	0.00018
Manuscript	4.3M	2.8M	0.00030	0.00026
Dragon	7.2M	3.6M	0.00052	0.00054
Statuette	10M	4.5M	0.0010	0.0015

Table 1 Example comparisons between a simplification satisfying our Gaussian sampling and QSlim. Errors are measured relative to the bounding box diagonal by sampling the Hausdorff distance between the original and the simplified meshes.

#faces	Approximation error	
	Our method	QSlim
52703	0.0035	0.0013
43924	0.0042	0.0014
35146	0.0049	0.0014
26369	0.0065	0.0017

Table 2 Comparison of the approximation error on the Armadillo model (345944 faces originally) using our method and QSlim. Each row is coarser than the previous, as indicated by the number of vertices. Errors are measured relative to the bounding box diagonal by sampling the Hausdorff distance between the original and the simplified meshes.

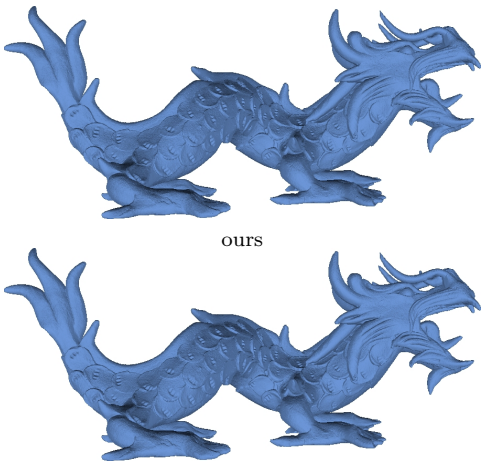

Fig. 16 Comparison to QSlim using an approximation reduced to 50% of the original size.

want to stress here that QSlim uses the quadric error metric [6] which directly approximates the $\mathcal{L}_2$ distance and thus optimizes for this metric in particular. Nevertheless, as we can see from Table 1, shape approximation based on our Gaussian sampling can achieve good simplification results for large models. Gaussian sampling targets another goal but still produces good simplifications also with respect to Hausdorff distance. A visual example is also given in Figure 16, where the Gaussian sampled and simplified model is compared to QSlim. As shown in Table 2, when the simplification is extreme, the optimal vertex placement of in-core methods like QSlim cannot be easily beaten. However, even under these unfavorable conditions, the error, measured against the bounding box diagonal, stays reasonably low, as illustrated in Figure 15. This result suggests the convenience of a combination of streaming and in-core methods when simplifying gigantic meshes.

However, the main advantage as mentioned before is the locality of our sampling method. The effectiveness in that respect can be demonstrated by the memory footprint, or the number of mesh elements, that have to be kept in main memory for sampling and mesh simplification. In Table 3 we show the results of the stream-processing implementation described in Section 5. As we can see, the memory requirements are very low as the maximum size of mesh elements, of the active set in the sliding window, is only a small fraction of the overall size of the processed models.

7 Conclusion and Future Work

We have presented a novel surface sampling technique derived from the imposition of three reasonable sampling conditions: anisotropy, sufficiency and minimality. To meet these conditions, we create the ϵ-net sampling in the Euclidean distance metric on the Gaussian sphere, which translates to an ϵ-net sampling in the $\mathcal{L}_{2,1}$ metric on the given surface. The resulting sam-

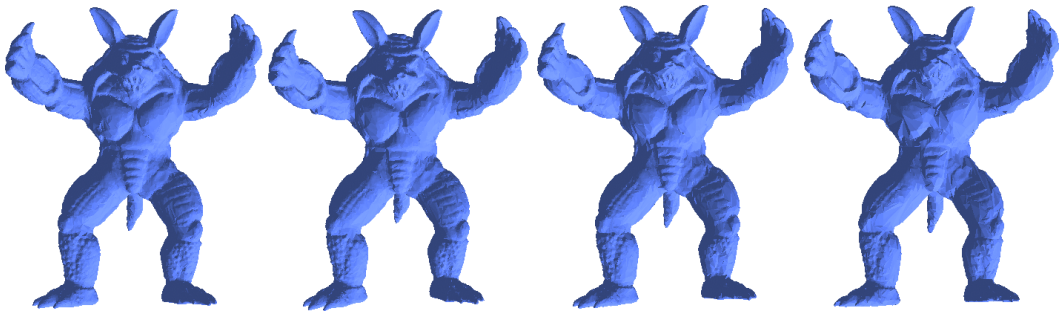

Fig. 15 Sequence of simplifications of the armadillo model, with 15%, 12%, 10%, and 7% of the original vertices, left to right.

Model	Max. Active Set Size	
	2 subdivisions	4 subdivisions
Armadillo	43016	49480
Manuscript	91893	63702
Dragon	394324	467161
Statuette	418581	495466

Table 3 Maximal size of the active set during stream-processing various meshes, given for a Gaussian sampling with sphere subdivision 2 and 4 respectively.

pling size is proportional to the total absolute Gaussian curvature of the given shape. This technique is the basis for a robust shape approximation algorithm with guarantees on topological correctness by construction, and it is amenable to streaming implementation. Further, although shape approximation through Gaussian sampling aims at bounding the normal deviation, it still produces good simplifications with respect to the Hausdorff distance. One promising direction for future work is non-uniform Gaussian sampling. A changing Gaussian sampling density can be used to represent a notion of importance sampling, with applications in rendering and importance driven approximation.

Acknowledgements We would like to thank Behzad Sajadi for his help preparing some of the images in this paper. We also thank the Digital Michelangelo Project at Stanford University and AIM@SHAPE for the provided models. This work was partially supported by the NSF grants IIS-0712253, CCF-0738401 and CCF-0811809, and the Swiss National Science Foundation Grant 200021-111746/1.

References

1. Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., Taubin, G.: The ball-pivoting algorithm for surface reconstruction. IEEE Vis. **5**(4) (1999)
2. Clarkson, K.L.: Building triangulations using ϵ-nets. In: SIGACT Symposium. ACM SIGACT (2006)
3. Cohen-Steiner, D., Alliez, P., Desbrun, M.: Variational shape approximation. ACM ToG. **23**(3), 905–914 (2004). DOI http://doi.acm.org/10.1145/1015706.1015817
4. Dey, T.K.: Curve and Surface Reconstruction: Algorithms with Mathematical Analysis (Cambridge Monographs on Applied and Computational Mathematics). Cambridge University Press, New York, NY, USA (2006)
5. Dong, W., Li, J., Kuo, J.: Fast mesh simplification for progressive transmission. In: Proceedings International Conference on Multimedia and Expo ICME 2000. IEEE (2000)
6. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: Proceedings ACM SIGGRAPH, pp. 209–216. ACM SIGGRAPH (1997)
7. Gopi, M., Krishnan, S., Silva, C.: Surface Reconstruction using Lower Dimensional Localized Delaunay Triangulation. Eurographics **19**(3), 467–478 (2000)
8. Gruber, P.M.: Optimum quantization and its applications. Adv. Math. **186**, 456–497 (2004)
9. Heckbert, P.S., Garland, M.: Survey of polygonal surface simplification algorithms. Tech. rep., Computer Science Department, Carnegie Mellon University (1997)
10. Hoppe, H.: Progressive meshes. In: SIGGRAPH, pp. 99–108. ACM SIGGRAPH (1996)
11. Isenburg, M., Lindstrom, P.: Streaming meshes. In: Proceedings IEEE Visualization, pp. 231–238 (2005)
12. Luebke, D.P.: A developer's survey of polygonal simplification algorithms. IEEE Comput. Graph. Appl. **21**(3), 24–35 (2001)
13. Nadler, E.: Piecewise-Linear Best l_2 Approximation on Triangulations. In: C.K. Chui, L.L. Schumaker, J.D. Ward (eds.) Approximation Theory V, pp. 499–502. Academic Press (1986)
14. O'Neill, B.: Elementary Differential Geometry, 2nd edn. Academic Press, www.apnet.com (1997)
15. Pajarola, R.: Stream-processing points. In: Proceedings IEEE Visualization, pp. 239–246 (2005)
16. Pottmann, H., Krasauskas, R., Hamann, B., Joy, K., Seibold, W.: On Piecewise Linear Approximation of Quadratic Functions. Journal of Geom. Graphics **4**(1), 31–53 (2000)
17. Sheffer, A.: Model simplification for meshing using face clustering. CAD **33**, 925–934 (2001)
18. Vetterli, M., Marziliano, P., Blu, T.: Sampling signals with finite rate of innovation. IEEE Transactions on Signal Processing **50**(6), 1417–1428 (2002). DOI 10.1109/TSP.2002.1003065
19. Weiler, K.: Edge-based data structures for solid modeling in curved-surface environments. IEEE Computer Graphics and Applications **5**(1), 21–40 (1985)
20. Yamauchi, H., Gumhold, S., Zayer, R., Seidel, H.P.: Mesh segmentation driven by gaussian curvature. The Visual Computer **21**(8-10), 649–658 (2005)