EUROGRAPHICS 2008 / G. Drettakis and R. Scopigno
(Guest Editors)

Volume 27 (2008), Number 3

Memoryless Gauss Mesh Sampling

Paper ID: 44887

Abstract

Studying the relationship between Gaussian space and a surface can provide invaluable information about the properties of the surface. In this paper, we use recent results on quantization and surface approximation theory to propose a simple, robust, linear time, feature preserving mesh sampling algorithm that can be easily controlled by the user. This algorithm is based on the explicit inverse one-to-many mapping of a regular sampling of the Gaussian sphere onto a manifold surface, and it is memoryless in the sense that most of its operations are local and no global information is maintained. For the same reason, this algorithm can also be run in parallel without dependencies between the samples. We demonstrate our sampling method by applying it to the problem of shape approximation.

1. Introduction

Sampling 3D models involves choosing points from the surface such that an interpolation of these points faithfully reproduces the desired features of the given model, both in terms of geometry and topology. Mesh sampling is important in many geometry processing problems including shape approximation, surface reconstruction and parameterization. In this paper, we propose a linear time algorithm to produce a feature sensitive sampling of a surface. The output sample set is a subset of the vertices of the input mesh, and can be shown to be minimal for a given sampling error.

The fundamental challenge in surface sampling is to find the minimal set of sample points that capture the features of a shape within an (approximation or sampling) error bound. For shape approximation, [Cla06] proved that the sampling size is proportional to the integral of the absolute Gaussian curvature over the surface. This result has been independently proved for surface approximation [CSAD04] and surface reconstruction [ACK01, Eri01], with different interpretations of “Gaussian curvature”. In the surface reconstruction literature, the interpretation of the Gaussian curvature at a point on the surface is the square of the curvature of the medial axis ball tangential to that point. We use this observation to directly approximate the total absolute Gaussian curvature by regular sampling of the Gaussian sphere. We thus find the appropriate surface samples by inverse mapping these regular Gaussian samples back onto the surface (a one-to-many mapping). The result is a simple and robust linear time surface sampling algorithm. This sampling algorithm has potential uses in shape approximation, topology processing, streaming and out-of-core simplification of large data sets, as well as parameterization and compression of meshes. We demonstrate the applicability of our sampling algorithm as the first step of a shape approximation technique.

1.1. Main Contributions

Following are the main contributions of this paper:

- We introduce the sampling of a surface as on the one-to-many mapping of a sampling on Gaussian space back onto the surface.
- We prove that the above sampling set is minimal with respect to the given sampling error for a class of models.
- We present a practical algorithm that applies the above theory for sampling polygonal meshes. This method is very efficient and hence the sampling can even be interactively driven by the user.
- We extend the sampling method to solve 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 in Section 5.

2. Previous Work

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

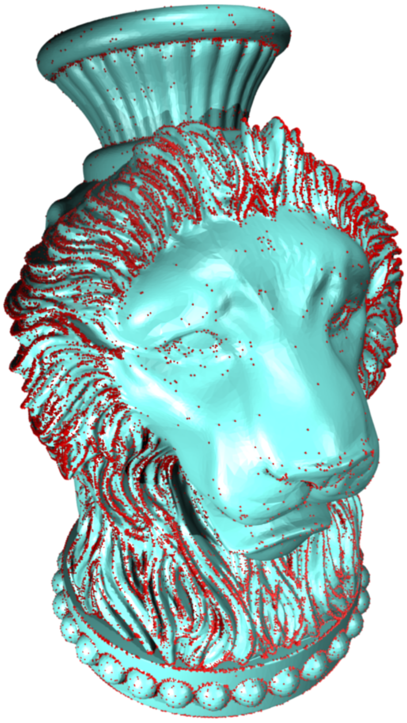

Figure 1. Feature sensitive samples on a mesh.

Gaussian Sampling: We broadly classify the algorithms that use normal vectors for geometry processing as Gaussian sampling algorithms. In [JC01] face normals are clustered in order to quickly extract silhouettes for shadow boundary computation. In order to precompute radiance transfer, [SHHS03] accumulates surface samples with similar orientation. Convergence of the iterative closest point algorithm can be accelerated by selecting sample points according to the normal variation they introduce in the shape [RL01]. Other applications of Gaussian maps include partitioning of mesh into developable surfaces [JKS05] and mesh segmentation [YGZS].

Shape Approximation: Surface simplification or shape approximation techniques aim at reproducing a given surface with minimum error using fewer mesh elements than in the original. The vertices of this approximation can be considered a feature sensitive sampling of the original surface, which indicates their relationship with our problem. In this context we would like to distinguish between *sampling* and *approximation* error. While *sampling* error is used to evaluate the effectiveness of a sampling technique (distance among samples, and between any point on the surface to its closest sample), *approximation* error is used in the context of approximation algorithms (distance between the original and the simplified surfaces).

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. Since there is a long history of surface simplification algorithms, we refer to excellent surveys in this field [HG97, Lue01]. In general, these methods try to optimize an energy functional or iterate in order to find (optimal) positions and shape of the mesh elements (vertices, edges and faces) that would reduce the *approximation error* [CSAD04, She01].

3. Sampling of Smooth Manifolds

Following are three desirable properties for a sampling S on a surface U:

1. **Anisotropy**: At any point on the surface, sampling should be denser along the lower curvature direction than along the higher curvature direction.
2. **Sufficiency**: All points $x \in$ U should not be further than an ε distance from their closest sample.
3. **Minimality**: Every sample $s \in$ S should be necessary; In other words, removing it would make S violate some of the previous conditions.

The remainder of this section is devoted to explaining how our sampling method meets the stated conditions. For the purpose of sampling, we measure the distance between two points x and y on the surface as the Euclidean distance between the normal vectors at x and y (written $\mathbf{n}(x)$ and $\mathbf{n}(y)$), or $D(x,y) = ||\mathbf{n}(x) - \mathbf{n}(y)||$. This metric, also known as the $\mathcal{L}_{2,1}$ metric, naturally captures the *anisotropy* of the model as shown by [PSH*04, CSAD04].

Definition 1. ***ε-covering:*** *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 *that is at distance not greater than* ε ($\mathcal{L}_{2,1}(x,s) \leq \varepsilon$)*, then the sampling* S *is called an* ε*-covering of the surface* U.

An ε-covering S of surface U meets the condition of *sufficiency* by definition. Moreover, a sample set T of surface U is an ε-covering of U if it contains a sample set S which is also an ε-covering of U. Although finding the smallest of all possible ε-coverings is NP-hard, in order to just meet the condition of *minimality* we can start with a sufficient sampling and simply remove all the unnecessary samples. One way to get fewer samples is to force a minimum distance between samples.

Definition 2. ***ε-packing:*** *Let* S *be a set of samples on surface* U, *and* s, t *be two adjacent* [*] *samples,* $s,t \in$ S. *If* $\mathcal{L}_{2,1}(s,t) \geq \varepsilon$, *then the sampling* S *is called an* ε*-packing of the surface* U.

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

[*] Let the distance between s and t be d. We say two samples s,t are *adjacent* if there is a connected path between s and t on the surface such that the distance between any point on this path to s (and t) is less than or equal to d.

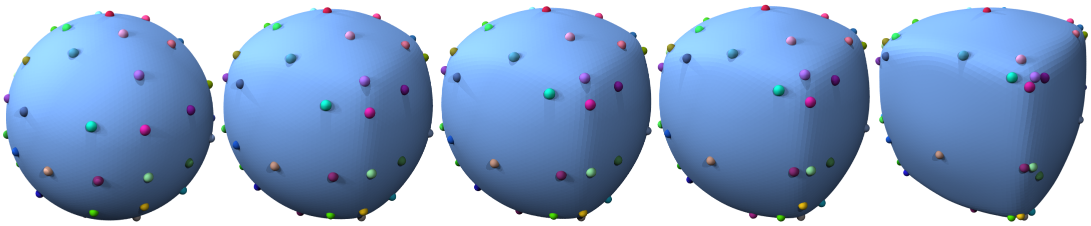

Figure 2. As a sphere morphs into a rounded cube, samples migrate towards high curvature regions. Since all the objects in this sequence are convex, genus zero manifolds, the total number of samples is constant.

Given the above definitions of sampling, we can derive a lower bound on the number of samples required to sample a smooth surface with the given error. Gruber [Gru04] and later, in the context of ε-net, Clarkson [Cla06] prove that the sample size of an ε-net sampling of a surface U using the distance metric $\mathcal{L}_{2,1}$ is proportional to $\int_U |K(x)|dx/\varepsilon^2$, where $K(x)$ is the Gaussian curvature at the point x. In other words, *the sample size is proportional to the total absolute Gaussian curvature of the surface.* A similar result on the sample size for surface reconstruction algorithm was proved by [Eri01].

From the above arguments, we conclude that an ε-net sampling of $\mathcal{L}_{2,1}$ on the surface U would satisfy our desiderata, and this constitutes the basis for our sampling algorithm.

3.1. 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 surface U is equivalent to the 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. (For practical reasons, a tetrahedron, octahedron or icosahedron are preferred.) Let l be the distance between any two adjacent vertices of this uniform tessellation. The vertices are at least $l-\mu$ away from each other (μ is a small positive number), and any point on the sphere is less than $l-\mu$ distance from its closest vertex in the tessellation. In other words, the vertices of this tessellation form an $(l-\mu)$-net sampling of the Euclidean metric of the sphere.

Given the vertices of the uniform tessellation of a unit (Gaussian) sphere, *the ε-net sampling of $\mathcal{L}_{2,1}$ on the surface U is given by the sample set S consisting of all those points x on U whose normal vector is one of the vertices of the Gaussian sphere.* Different orientations of the Gaussian sphere will produce different samplings on U but all of these sample sets are equivalent, in the sense that they satisfy the same sampling properties.

3.2. Properties of Our Sampling

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 [O'N97]). Using the following definition of convexity, the previous result can be extended to 2-manifolds with higher-genus, considering objects like a torus to be convex for our purposes.

Definition 4. ***Manifold n-convexity****: A 1-manifold (closed curve) M_1 on a plane is convex (1-convex) if and only if the line segment between any two points in its interior does not intersect M_1. A compact 2-manifold M_2 with arbitrary genus is said to be 2-convex if and only if the intersection of any plane with M_2 is a set of 1-convex 1-manifolds.*

Lemma 1. *The total absolute Gaussian curvature of a genus g, 2-convex 2-manifold equals $4(g+1)\pi$.*

Since the total absolute Gaussian curvature is the same for any 2-convex model with genus g, regardless of any other shape considerations, the total number of samples should also be the same for all these models (for a given tessellation of the Gaussian sphere). Also, the sampling produced by our method is minimal for this class of models. Proof of minimality is provided in the Appendix.

Figure 2 illustrates this property. We show a sequence of models depicting the transformation of a sphere into a rounded cube. As expected, the number of samples remains the same. Moreover, the distribution of these samples exhibits anisotropy, with more samples migrating towards the high curvature regions. The normal deviation between the adjacent samples is constant by construction.

The properties of the application of our sampling method to higher genus objects is illustrated in Figure 3, which shows the samples obtained on two (convex) torii of different sizes. In both cases, the number 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 Lemma 1, 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 (like those in Figure 2).

Finally, Figure 4 illustrates the effect of concavities in the sampling size. The total absolute Gaussian curvature of the

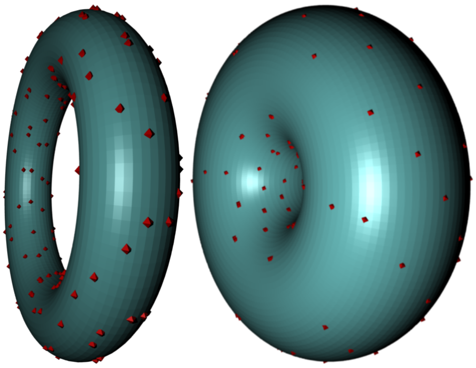

Figure 3. The sampling size produced by our algorithm is proportional to the total absolute Gaussian curvature. For a 2-manifold, this value is constant, regardless of size and shape, as long as it is 2-convex. So the total number of samples generated is the same for both the two torii above.

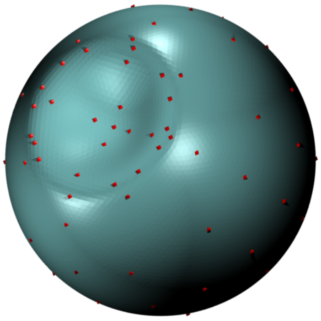

Figure 4. A concavity in the shape increases its total absolute Gaussian curvature. Correctly sampling it requires more samples than a convex shape.

model shown is higher than that of the convex objects from Figure 2, and therefore more samples are produced to maintain the proportionality.

Sampling for Shape Approximation: An ε-covering S of a slightly perturbed version of the $\mathcal{L}_{2,1}$ metric has been shown to exhibit Hausdorff bounds on the error between the original surface U and the triangulation constructed using the samples in S [Cla06]. The topological correctness of the approximation must be enforced by appropriately connecting the samples in the same way as they are connected in the original surface.

4. Sampling Manifold Meshes

In this section, we present an algorithm that applies the above theory to the problem of sampling polygon meshes, including those with sharp features. We suggest an initial numerical approach and a second, more robust approach, based on the discretization of the space of normals.

For a *smooth* manifold M, our samples are those points on M whose normal matches exactly with one of the vertices of the tessellation of the Gaussian sphere G. For a piecewise linear approximation of M, such as a triangle mesh, the probability of a vertex normal coinciding exactly with a Gaussian vertex is zero. In a triangle mesh, normals change only at the edges and vertices. 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 triangles incident on v. The mesh vertex v is considered a sample if and only if its spherical polygon on the Gaussian sphere G contains one or more vertices of G. This procedure is described in Figure 5.

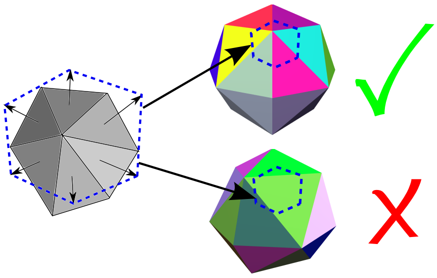

Figure 5. In the ideal conception of our algorithm, a mesh vertex is selected as a sample if the spherical polygon induced by the normals of its incident faces encloses a Gaussian vertex.

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 of this numerical approach we proceed by quantizing the normal vectors as described below.

Definition 5. ***Triangle association, feature edge, candidate sample:*** *A mesh triangle t with normal n_t is* associated *to a Gaussian triangle t_G if n_t pierces t_G in normal space. An edge e of a triangle mesh is a* feature edge *if the two incident faces are associated to different Gaussian triangles. A vertex v of a triangle mesh is a* candidate sample *if three or more* feature edges *are incident on it.*

Let us quantize the normal vectors such that the normal vector of a triangle in the mesh is assigned the normal vector of its associated Gaussian triangle. Under this reassignment, it can be seen that our sampling set is a *subset* of all the candidate samples in the mesh, given a specific tessellation G of the Gaussian sphere. We discard those which are not samples by applying three filtering rules: untriangulatable patch filtering, non-sample filtering and noisy sample filtering.

Untriangulatable Patch Filtering: We observe that feature edges form closed curves, thus partitioning the input mesh. Triangles within a partition are associated to the same Gaussian triangle. In order to approximate the shape of a patch, it should have at least three samples in its boundary. But due to noise in the mesh and normal vectors of low curvature mesh regions that are roughly aligned with a Gaussian

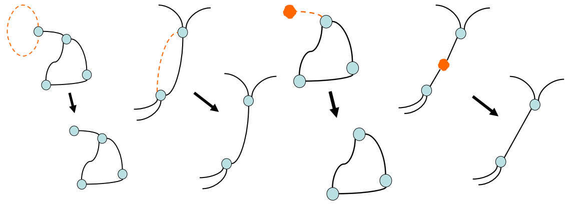

Figure 6. Four cases of degenerate partition boundaries and how they are cleansed before constructing the polygons. From left to right: Polygon boundary with (a) one or (b) two feature vertices; Former candidate sample with only (c) one or (d) two incident feature edges.

vertex, it is quite possible that a few patches are not triangulatable. Figure 6 shows these degenerate conditions and how we process them. Note that Figure 6 cases (c) and (d) do not occur naturally. Resolving case (a) may lead to case (c), and case (b) may lead to case (d).

Non-sample Filtering: 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 triangles incident on a mesh vertex v. Let A_1 and A_2 be associated to a, B_1 and B_2 be associated to 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 triangles around v does not enclose any Gaussian vertex and hence v cannot be a sample. Such cases can be generalized as follows. Consider the spherical polygon formed by the normal vectors of the incident triangles on the mesh vertex v. If consecutive vertices of this spherical polygon fall in adjacent (edge connected) Gaussian triangles, and if these Gaussian triangles form a tree (a unique path exists within this set of Gaussian triangles to go from one triangle to another), then that spherical polygon cannot enclose a Gaussian vertex, and hence v cannot be a sample. See Figure 8 for an example.

Noisy Sample Filtering: In noisy and low curvature regions of the mesh, there would still be superfluous samples that are not eliminated by the above two filtering techniques (Figure 9). One of such cases occurs when multiple candidate samples representing the same Gaussian vertex exist in the same local neighborhood, even though clearly a subset of them would suffice to represent that region (and that Gaussian vertex). The noisy-sample filtering technique detailed here identifies the samples that have to be retained; the rest of the candidates can be removed.

Represented Gaussian Vertex: A candidate sample v represents a Gaussian vertex v_G if all the associated Gaussian triangles of the mesh triangles incident on v are incident on v_G.
Convex, Concave, and Saddle Samples: A candidate sample v is convex (or concave) if for counter-clockwise traversal of the incident triangles around v, the spherical polygon on the Gaussian sphere formed by the normal vectors of the incident triangles enclose the Gaussian vertex in counter-clockwise (respectively, clockwise) direction. It is a saddle vertex if the spherical polygon is self-intersecting. We call such a classification of a candidate sample as its *curvature characteristic*.
Adjacent Samples: Two candidate samples v_1 and v_2 are adjacent to each other on the mesh if there is a sequence of feature edges connecting v_1 and v_2 that does not pass through any other candidate sample.

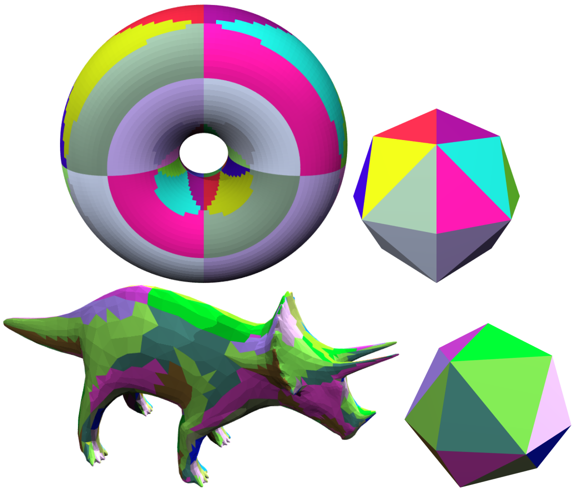

Figure 7. The faces of two meshes (left) are clustered according to their associated Gaussian triangles (Gaussian sphere shown to the right). These clusters are separated by *feature edges*. Vertices adjacent to three or more *feature edges* are *candidate samples*.

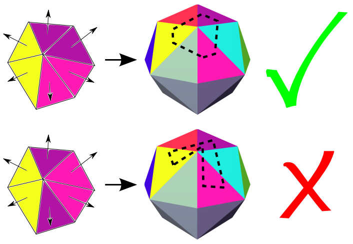

Figure 8. Candidate samples (left) can be discarded if the normals of their incident faces form a spherical polygon (middle) that cannot contain a Gaussian sample.

As the first rule of retaining samples, we retain two adjacent candidate samples representing the same Gaussian vertex if they have different curvature characteristics. The second rule of retaining samples is as follows: Among a group of adjacent candidate samples which represent the same Gaussian vertex and have the same curvature characteristic, a subset of them has to be retained such that the union of their associated triangles covers all the incident Gaussian triangles of the Gaussian vertex (see Figure 10). Finding the

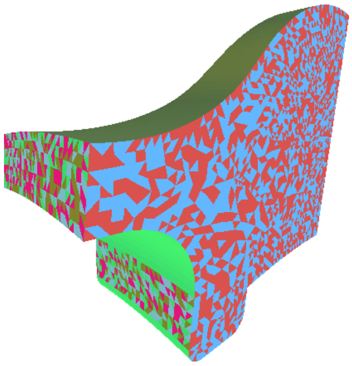

Figure 9. Low curvature regions which are roughly aligned with a Gaussian edge or vertex produce an excess of candidate samples.

minimum number of such candidate samples from the set is equivalent to the min-vertex-cover graph problem. Although good approximation algorithms exist for this NP-complete problem, since the number of Gaussian triangles incident on a Gaussian vertex is very small, we can even go for an exhaustive search for the min-cover solution. Any candidate sample not retained by either of the above two rules can be safely removed as a noisy-sample. Figures 1, 11, 12 and 13 contain examples of sampling results.

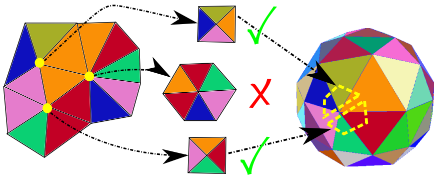

Figure 10. Some candidate vertices (yellow circles) can be discarded if they are adjacent to other candidate vertices representing the same Gaussian vertex and with the same curvature characteristic. A subset of these vertices is retained such that the union of their incident triangles covers all the incident Gaussian triangles around the represented Gaussian vertex. Here, the middle vertex is redundant, because its incident triangles are associated to Gaussian triangles that are covered by its neighbors.

«««< sampling.tex

4.1. Effects of Noise in the Mesh

Consider an almost planar region of the mesh whose normal coincides with a Gaussian vertex v_G. With minor noise in the mesh, the normal vectors of the triangles in the planar region will be associated to different Gaussian triangles around v_G, and hence many mesh vertices will be considered *feature vertices*. Extensive studies in computational geometry point out that such cases constitute degeneracies [EM90]. By the definition of degeneracy, they can be removed by a slight perturbation of the orientation of the Gaussian sphere.

On the other hand, consider a smooth curved region of the mesh whose normal space span a large region in the Gaussian sphere (as in the back of the horse in Figure 14). If such a region has noise, then it cannot be removed by perturbation of the orientation of the Gaussian sphere. We have to handle the noise in the normal vectors directly by applying the low-pass Laplacian filter [Tau01] on them. In our experiments (see Figure 14), we observed that a single iteration of Laplacian smoothing is usually enough to remove all the irrelevant concavities and the samples.

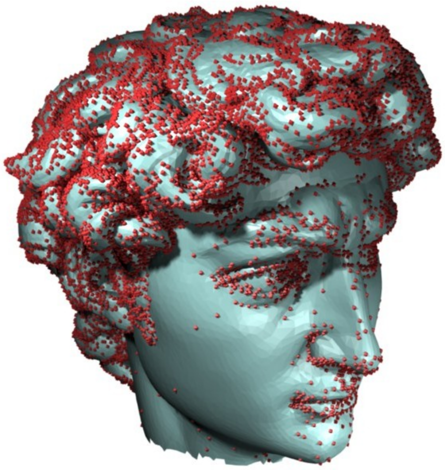

Figure 11. Feature sensitive samples produced by our algorithm on a large mesh. The Gaussian sphere was tessellated using 112 triangles.

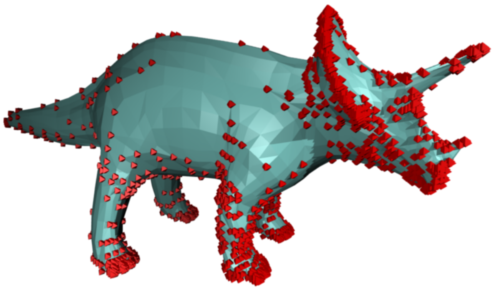

Figure 12. Samples produced by our algorithm on a mesh. Notice the dramatic changes in sample distribution between low and high curvature regions.

======= »»»> 1.34

Memoryless Sampling: Almost all the presented operations, such as finding a candidate sample and non-sample filtering, are local operations around the mesh vertices. Hence the entire sampling algorithm requires no memory or communication/sharing of data and thus is embarrassingly parallelizable on each mesh vertex. If an application can accept more than a minimal set of samples, the candidate samples chosen by the above operations would well serve the purpose. The other two operations, namely the untriangulatable patch filtering and noisy-sample filter require limited local

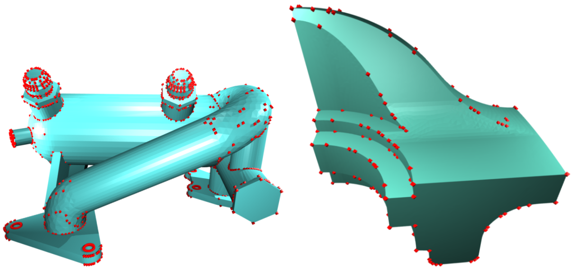

Figure 13. Samples produced on two mechanical parts.

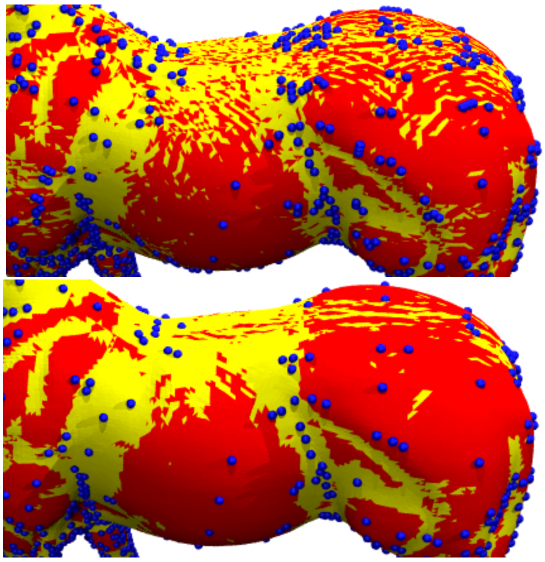

Figure 14. Concave regions are shown in yellow and convex in red. Blue spheres represent samples. A low-pass filter on the mesh normals reduces the noise in the normal variation of the original mesh (top), eliminating spurious concavities and reducing the number of samples in the output (bottom).

traversal of the incident feature edges, but can still be performed with minimal memory using appropriate indexing and storage of mesh triangles based on their normal vectors.

5. Shape approximation

By construction, the *feature edges* of a mesh form closed curves on the surface, which partition the mesh into clusters of connected triangles *associated* to the same Gaussian triangle (see Section 4). This implies that these clusters only contain faces with bounded normal deviation and it must therefore be possible to substitute each such cluster with a set of similarly oriented triangles (bounded normal deviation). Illustrations of this partitioning are shown in Figure 7. We can therefore approximate each of these patches using a simpler proxy.

In order to produce a triangulated approximation to the original shape, two approaches can be considered. The first and most straightforward one produces a triangulation using only the vertices marked as samples. To do so, we compute the Voronoi diagram of the original shape using the sample vertices as Voronoi centers, and the $\mathcal{L}_{2,1}$ metric. As is done in [CSAD04], when we encounter a mesh triangle whose three vertices belong to different Voronoi cells, we output a triangle connecting the Voronoi centers of those three cells.

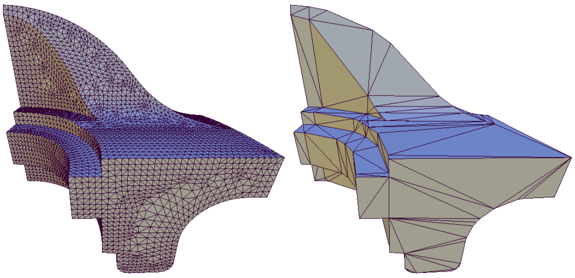

Figure 15. Shape approximation on a mechanical part. The triangle edges are rendered in black for clarity.

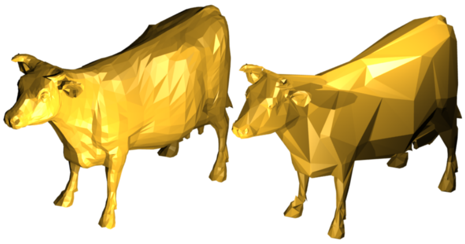

Figure 16. A cow model 5830 faces (left) approximated using 1996 faces (right).

The second approach involves applying the above algorithm on a per-patch basis, using the corner vertices of each patch (this is all the original candidate samples, even those not selected as samples) as Voronoi centers. Because not all the vertices used in this triangulation are samples, a series of edge collapse operations may be applied as necessary. The added complexity of this second approach offers the guarantee of topological correctness. Furthermore, since each patch is relatively small, any geometric discrepancies with respect to the original model can be solved at low cost by rearranging the produced triangles. Both triangulation algorithms run in $O(n \log n)$ time on the size of the original mesh. Some results of the approximation algorithm are shown in Figure 15, 16 and 17.

Effect of the Orientation of the Gaussian Sphere: In our algorithm, the sampling on the surface changes by different orientations of the Gaussian sphere. But both the sampling and the approximation error bounds of the surface approximations given by any of these samplings are the same, since that bound depends only on the density of the Gaussian sphere tessellation. This is demonstrated in Figure 18 in which the Hausdorff error between the original model and the simplified model generated by the samples from different orientations of the Gaussian sphere are plotted. Although the

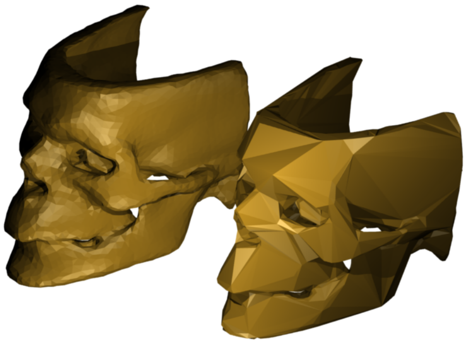

Figure 17. A skull model with 22104 faces (left) approximated using 6462 faces (right).

error is maintained for different orientations of the sphere, the number of samples might be different in each of these cases. Given an error bound, finding the optimal orientation of the Gaussian sphere tessellation in order to minimize the sample size is NP-hard, as can be derived from [CVM*96]. On the positive side, identification of *feature vertices* in the mesh is so fast that the user can interactively select a good orientation of the Gaussian sphere for (visually) better sampling.

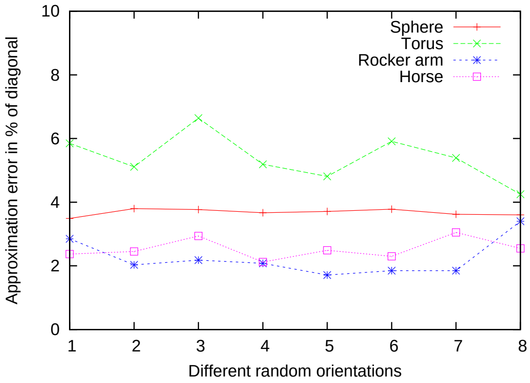

Figure 18. Approximation error for different input meshes and eight random orientations of the tessellated Gaussian sphere. The error, measured as a percentage of the size of the model, is never greater than 8%.

Handling Meshes with Boundaries: The holes of the mesh are treated like special mesh partitions themselves. If the boundary of a hole has at least three *feature* vertices, a *hole polygon* is created, and the hole remains after triangulation. Otherwise the hole is eliminated as its associated polygon is degenerate. Interestingly, while boundary curves on the mesh which are close to geodesics are approximated automatically (Figure 19), other boundary curves whose Frenet frame normal vector is perpendicular to the mesh normal are filled by default, but this can be avoided with user intervention.

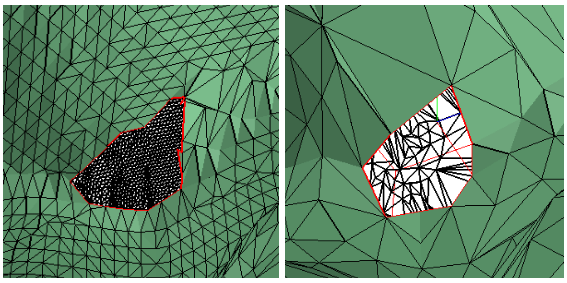

Figure 19. By tagging the boundaries of the model as feature edges for the approximation, we can ensure that they are retained in the simplified output. In the figures above, a hole in the chest of the bunny model remains after simplification and is well approximated by our algorithm.

6. Application: Segmentation for Spherical Parameterization

Various techniques have been developed for segmenting topologically complex polygonal meshes into patches homeomorphic to a disk. Not many attempts have been made to segment meshes into patches homeomorphic to other well known topological entities, like the sphere or torus. These domains have various advantages over the disk including a naturally seamless continuity of the parameterization. In this section we present a technique to partition the mesh into patches that are directly parameterizable using spherical coordinates.

Overview of the algorithm: For a convex object with genus g, the normal vector space would wrap the Gaussian sphere $g+1$ number of times (and hence the Lemma 1 which states that the total absolute Gaussian curvature is $g+1$ times the area of the unit sphere). The fundamental idea behind mesh segmentation into spherical pieces is to segment the mesh in such a way that the normal vector space of each contiguous partition would wrap the sphere exactly once. Thus we will get exactly $g+1$ partitions from a genus g convex object.

In non-convex objects, the normal vectors of more than $g+1$ points of the object might map to the same point of the Gaussian sphere. If we perform a low pass filtering on the normal vector space sufficient number of times, then the number of pre-images of all points (except for a finite number of points) of the Gaussian sphere would be exactly $g+1$. The effect of the low pass filtering on the normal space is to remove the concavities of the model and make it convex. The finite number of points on the Gaussian sphere which has fewer than $g+1$ pre-images on the model are called *poles*. If we remove the pre-imaegs of the poles from the model, the rest of the model will be automatically separated into $g+1$ pieces. The pre-images of the poles are then attached to the closest partition of the model to create $g+1$ patches that are directly parameterizable using spherical coordinates.

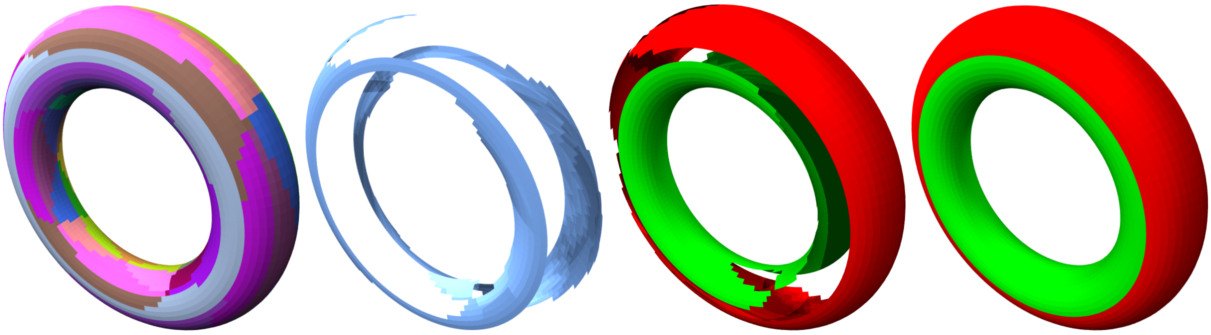

Figure 20. Segmentation process of a torus, from left to right. **First:** Normal based partitioning of the input mesh. **Second:** Patches associated to a Gaussian triangle that appears less often than the others are marked as poles. **Third:** Non-pole patches can be directly aggregated, forming the skeleton of the segmentation. **Fourth:** Final segmentation, after assigning all pole faces to adjacent segments.

6.1. Algorithm for Mesh Segmentation

Given the tessellation of the Gaussian sphere, each partition of a polygonal mesh as done in Section 4 is a pre-image of a Gaussian triangle. The number of pre-images of a Gaussian triangle may be more than $g+1$. With sufficient smoothing of mesh normals [Tau01], the number of patches on the mesh associated with every Gaussian triangle can be brought down to no more than $g+1$. Note that, since all our processing is with the normal vectors, we do not change the positions of the vertices while smoothing the normal. The spherical-parameterizable segmentation we are looking for can be produced by finding $g+1$ connected sets of patches such that in every set each Gaussian triangle is not represented by more than one patch.

Patch grouping: Finding the connected set of unique patches as described above is equivalent to the following graph problem. Consider a colored undirected graph G using c different colors. We need to find a graph partitioning into connected subgraphs such that each subgraph has at most one vertex of each color. This problem is known to be NP-hard [SSKBD01], and therefore no efficient solution is believed to exist.

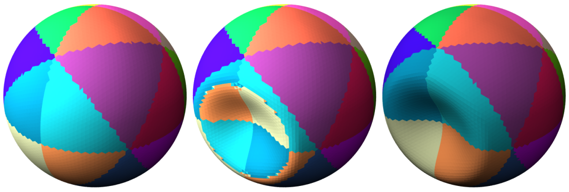

Figure 21. Left: In convex objects, each Gaussian triangle is represented exactly once. **Middle:** Concavities force some Gaussian triangle to be representated more than once. **Right:** By smoothing its normals, a concave object can be treated as a convex one.

We have found a heuristic that exploits the characteristics of the graph induced by the manifold mesh to produce a correct solution in much shorter time. The heuristic is based on the notion of *poles*, or patches with neighborhood not homeomorphic to a disk. Pole patches are identified by being associated to a Gaussian triangle that appears less than $(g+1)$ times in the mesh. These patches isolate the non-pole patches that form the g handles of the manifold. A greedy heuristic that exploits this notion is illustrated in Figure 20. The heuristic first aggregates all connected non-pole patches such that no aggregate has more than one patch associated with the same Gaussian triangle. Then triangles from pole patches are assigned to the closest adjacent non-poles, using the $\mathcal{L}_{2,1}$ metric. So long as the normals of the input object are smoothed (Figure 21) as to functionally remove all their concavities, this heuristic produces correct results in all our experiments (Figure 22).

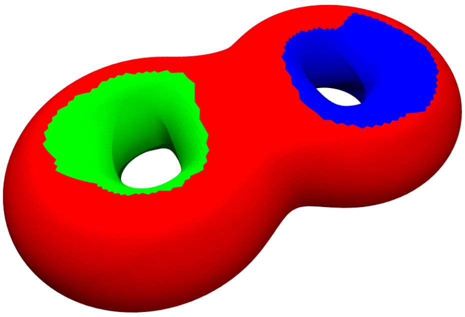

Figure 22. Color coded segmentation of a double torus into three patches parameterizable with spherical coordinates.

7. Conclusion and Future Work

We have presented a novel linear time surface sampling technique derived from the imposition of three reasonable sampling conditions. To meet these conditions, we create the ε-net sampling in the Euclidean distance metric on the Gaussian sphere, which translates to ε-net sampling in the $\mathcal{L}_{2,1}$ metric on the given surface. As shown before, the sampling size is proportional to the total absolute Gaussian curvature of the given shape. This sampling technique is the basis for a robust shape approximation algorithm with guarantees on topological correctness, by construction.

Two promising directions for future work include non-uniform Gaussian sampling and the parallel implementation of the method. In the former, a varying sampling density for the Gaussian sphere could be used to represent a notion of importance sampling, with applications, for example, in rendering. In the latter, the algorithm would be extended to handle different regions of large models concurrently, exploiting the fact that our sampling algorithm requires only access to local neighborhood in order to accept or reject a sample.

References

[ACK01] AMENTA N., CHOI S., KOLLURI R.: The power crust. In *Solid Modeling* (2001), pp. 249–260. 1

[Cla06] CLARKSON K. L.: Building triangulations using ε-nets. In *SIGACT Symposium* (2006). 1, 3, 4

[CSAD04] COHEN-STEINER D., ALLIEZ P., DESBRUN M.: Variational shape approximation. *ACM ToG. 23*, 3 (2004), 905–914. 1, 2, 7

[CVM*96] COHEN J., VARSHNEY A., MANOCHA D., TURK G., WEBER H., AGARWAL P., BROOKS F., WRIGHT W.: Simplification envelopes. In *SIGGRAPH* (New York, NY, USA, 1996), ACM Press, pp. 119–128. 8

[EHP02] ERICKSON J., HAR-PELED S.: Optimally cutting a surface into a disk. In *Sym. Comp. Geom.* (2002), pp. 244–253.

[EM90] EDELSBRUNNER H., MUCKE E. P.: Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM ToG 9*, 1 (1990), 66–104. 6

[Eri01] ERICKSON J.: Nice point sets can have nasty delaunay triangulations. In *Sym. Comp. Geom.* (2001), pp. 96–105. 1, 3

[Flo97] FLOATER M. S.: Parametrization and smooth approximation of surface triangulations. *Comput. Aided Geom. Des. 14*, 3 (1997), 231–250.

[Gru04] GRUBER P. M.: Optimum quantization and its applications. *Adv. Math. 186* (2004), 456–497. 3

[HG97] HECKBERT P. S., GARLAND M.: *Survey of Polygonal Surface Simplification Algorithms*. Tech. rep., 1997. 2

[JC01] JOHNSON D. E., COHEN E.: Spatialized normal come hierarchies. In *I3D* (2001), pp. 129–134. 2

[JKS05] JULIUS D., KRAEVOY V., SHEFFER A.: D-charts: Quasi-developable mesh segmentation. In *Eurographics* (2005), Eurographics, pp. 581–590. 2

[KLS03] KHODAKOVSKY A., LITKE N., SCHRÖDER P.: Globally smooth parameterizations with low distortion. In *SIGGRAPH* (2003), pp. 350–357.

[LPRM02] LÉVY B., PETITJEAN S., RAY N., MAILLOT J.: Least squares conformal maps for automatic texture atlas generation. In *SIGGRAPH* (2002), pp. 362–371.

[Lue01] LUEBKE D. P.: A developer's survey of polygonal simplification algorithms. *IEEE Comput. Graph. Appl. 21*, 3 (2001), 24–35. 2

[O'N97] O'NEILL B.: *Elementary Differential Geometry*, 2^{nd} ed. Academic Press, 1997. 3

[PSH*04] POTTMANN H., STEINER T., HOFER M., HAIDER C., HANBURY A.: The isophotic metric and its application to feature sensitive morphology on surfaces. In *ECCV (4)* (2004), pp. 560–572. 2

[PSS01] PRAUN E., SWELDENS W., SCHRÖDER P.: Consistent mesh parameterizations. In *SIGGRAPH* (2001), pp. 179–184.

[RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the ICP algorithm. In *Conf. on 3D Digital Imaging and Modeling* (2001), pp. 145–152. 2

[SCOGL02] SORKINE O., COHEN-OR D., GOLDENTHAL R., LISCHINSKI D.: Bounded-distortion piecewise mesh parameterization. In *IEEE VIS* (2002), pp. 355–362.

[She01] SHEFFER A.: Model simplification for meshing using face clustering. *CAD 33* (2001), 925–934. 2

[SHHS03] SLOAN P.-P., HALL J., HART J., SNYDER J.: Clustered principal components for precomputed radiance transfer. In *SIGGRAPH* (2003), pp. 382–391. 2

[SSKBD01] SINHA K., SUR-KOLAY S., BHATTACHARYA B. B., DASGUPTA P. S.: Partitioning routing area into zones with distinct pins. In *IEEE Conf. on VLSI Design* (2001), p. 345. 9

[SWG*03] SANDER P. V., WOOD Z. J., GORTLER S. J., SNYDER J., HOPPE H.: Multi-chart geometry images. In *SGP* (2003), pp. 146–155.

[Tau01] TAUBIN G.: *Linear Anisotropic Mesh Filtering*. Tech. rep., 2001. 6, 9

[YBS04] A fast and simple stretch-minimizing mesh parameterization. In *Shape Modeling Int.* (2004), pp. 200–208.

[YGZS] YAMAUCHI H., GUMHOLD S., ZAYER R., SEIDEL H.-P.: Mesh segmentation driven by gaussian curvature. *The Visual Computer 21*, 8-10, 649–658. 2

Appendix: Proof of Minimality The sampling produced by our method is *minimal* in the sense that any subset of the sample set is *not guaranteed* to be an ε-net sampling of the surface. In other words, there exists at least one object in which the sampling produced by our method is minimal.

Consider a regular tessellation of a Gaussian sphere in which the lengths of all the edges are l. An $(l-\mu)$-net sampling (where μ is a small positive quantity) of a convex genus zero object M will map the vertices of this Gaussian sphere to the points on M that has the same normal vectors. Let us assume that we remove one of these samples s from this sample set. Then we find that a point p $(= s)$ on M is more than

$(l-\mu)$ distance away (as calculated by the $\mathcal{L}_{2,1}$ metric) from any other sample in the sample set. Thus, $S-\{s\}$ is not an $(l-\mu)$-covering and hence not an $(l-\mu)$-net of M. Since the sample s is arbitrary, no subset of the sampling generated by our algorithm can be a valid $(l-\mu)$-net sampling of M. This result holds good for any convex 2-manifold with arbitrary genus.