Direct Sampling on Surfaces for High Quality Remeshing

Yan Fu* Bingfeng Zhou†
Institute of Computer Science and Technology, Peking University

Abstract

Isotropic point distribution is crucial in remeshing process to generate a high-quality mesh. In this paper, we present a novel algorithm of isotropic sampling on two-manifold mesh surface. Our main contribution lies in the successful generalization of a 2D fast Poisson disk sampling algorithm, which makes it able to directly sample 3D mesh surfaces, including feature edges. We adopt geodesic distance as the distance metric for sampling algorithm in 3D to better capture the geometry information. Given a density function over the surface, we derive a close analytic form of the available boundary, which makes our algorithm support efficient adaptive sampling. To further improve the isotropy of point distribution, Lloyd relaxation is performed locally to optimize the location of sampling points. The whole process guarantees that new vertices lie on the original surface. Mutual tessellation is utilized to reconstruct the connectivity of new vertices, which guarantees the fidelity and validity of topology. Experiments show that our algorithm is able to remesh an arbitrary closed manifold into a high-quality mesh with large minimal angles and small number of irregular vertices.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations;

Keywords: sampling, Poisson disk, surface remeshing

1 Introduction

Triangle mesh has been a prevalent form of 3D model representation in various areas ranging from modeling to visualization due to their simplicity and flexibility. In most applications such as *finite element analysis* and mesh editing [Yu et al. 2004], the quality of triangle meshes will greatly influence the stability and efficiency of numerical computations. High-quality meshes are also desirable in most existing geometry processing [Botsch et al. 2006]. Some input triangle meshes though capture shape accurately but have unsatisfactory quality. Hence, the *remeshing* process, which improves the quality of geometry and connectivity of original meshes, has been developed as a fundamental component of digital geometry processing.

The newly generated mesh after remeshing process should at least be valid and best approximate the original surface. As for mesh quality, the main concerned issues include vertex sampling, grading, regularity, size and triangle quality [Alliez et al. 2005]. In this work, we aim to provide a framework for *high quality remeshing* which traditionally means to generate a new triangulation from the input surface and the new mesh is required to possess well-shaped triangles, isotropic sampling and smooth grading. It is well known that isotropic vertex distribution can lead to well-shaped triangles and thus generate high-quality meshes. Therefore, isotropic sampling is crucial for applications where quality of the mesh elements is important.

Recent years have seen an explosion of research in surface resampling. Some of them apply uniform or adaptive sampling on planar parameter domains and then project the sampling points back to 3D surface, e.g. [Lee et al. 1998; Gu et al. 2002; Alliez et al. 2002; Alliez et al. 2003]. However, parameterization itself is challenging and always introduces severe distortions, especially for very complicated meshes [Floater and Hormann 2005]. Algorithms that directly sample on 3D surfaces usually involve a greedy process which gradually inserts new points based on Delaunay refinement [Boissonnat and Oudot 2003; Peyré and Cohen 2006] or a relaxation process that improves initial placement [Turk 1992; Vorsatz et al. 2003; Surazhsky et al. 2003]. Though the relaxation-based methods can achieve better results than the greedy approaches, they need lengthier computation and do not provide certified bounds on the triangle shape [Alliez et al. 2005]. Since a lot of algorithms that can generate isotropic sampling points on 2D domain have been proposed [Zhou and Fang 2003; Jones 2006; Dunbar and Humphreys 2006; Kopf et al. 2006; Ostromoukhov 2007], this leads us to extend 2D sampling algorithms to 3D surfaces. Previous work by Alliez et al. [2003] attempted to perform a discrete error diffusion process on triangle meshes, but achieved unsatisfactory sampling distribution. Recently, Dunbar and Humphreys [2006] described a modified Poisson-disk algorithm to generate isotropic point distribution with good spectral quality in $O(n)$ time. However, their method is currently restricted to generate uniform point sets in 2D domain.

In our work, we will demonstrate how to extend the method of [Dunbar and Humphreys 2006] to generate uniform / adaptive isotropic point distribution directly on 3D meshes. First, we adopt geodesic distance as the distance metric to precisely capture the geometry of 3D surface. Second, we derive a closed analytic form of available boundary for adaptive sampling. These two components are the kernel of our extension. As in 2D case, Lloyd iteration is then employed locally on 3D meshes to further improve the isotropy of point distribution. Our algorithm integrates the advantages of both greedy sampling and relaxation-based approaches: the extended Poisson-disk sampling algorithm is expected to produce nearly isotropic point distribution, which is verified by our experimental results ; with a good initial placement provided, only several iterations of local Lloyd relaxation can improve the initial point set into a precise isotropic distribution, which accelerates the convergence. When combined with some connectivity optimization techniques, our remeshing scheme can generate high-quality new meshes for arbitrary closed manifolds. In addition, our sampling algorithm supports level of detail and can be applied to object distribution on 3D surface.

We start by giving a brief overview of related work on resampling 3D mesh surfaces in section 2. Section 3 gives some preliminaries of our work. In section 4 we first present the algorithm for uniform sampling and then extend it to adaptive sampling in section 5. Next we describe the method for sampling points relocation in section 6. After presenting some experimental results in section 7, we conclude and mention some topics of possible future research

*e-mail: fuyan@icst.pku.edu.cn
†e-mail: cczbf@pku.edu.cn

in section 8.

2 Related Work

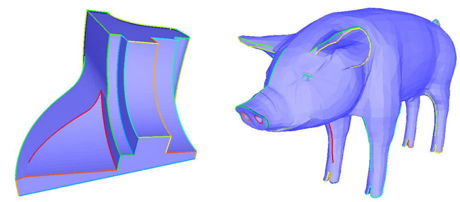
In recent years, numerous remeshing algorithms have been reported in the literature. The remeshing techniques can roughly be divided into two categories: the techniques relying on a parameter domain [Lee et al. 1998; Gu et al. 2002; Alliez et al. 2002] and those working directly on the 3D meshes [Boissonnat and Oudot 2003; Schreiner et al. 2006]. Compared with parametrization-based approaches, the second category avoids the tough mesh partition process and alleviates the distortion introduced by parametrization, also there is no need to handle the seams introduced by partition. Since our algorithm falls into the second category, our review of related work will focus on this field. For an excellent overview we refer the readers to [Alliez et al. 2005].

The first type of algorithms performed directly on 3D inserts one point at a time to refine the newly generated model greedily. Boissonnat and Oudot [2003] carried forward a "furthest point" strategy to progressively insert a new point at the center of the biggest void and update the 3D Delaunay triangulation simultaneously. This algorithm was later improved by Cheng et al. [2004] to achieve better efficiency. Since the sampling rules of [Boissonnat and Oudot 2003] and [Cheng et al. 2004] are both based on Delaunay / Voronoi geometry with Euclidean distance metric, triangles with small minimal angle may still exist in the final Delaunay triangulation results. The avoidance of such situations requires better selection of the location of sampling points. Based on front propagation with geodesic distance metric, Peyré and Cohen [2006] extended the furthest point sampling algorithm to achieve better results. Since this is a greedy algorithm, all the geodesic distance maps should be updated each time a new vertex is inserted, which makes the algorithm less efficient.

The main alternative to greedy sampling is relaxation-based methods, which initially place a point set on the surface and then improve the placement through point relocation. Turk [1992] proposed to apply an attraction-repulsion particle relaxation procedure on the initial placements. By adjusting the force between particles, vertices of different densities can be sampled on the surface. However, the time used for computing the solution of the diffusion equation makes it less efficient. Local area equalization is another efficient and robust way to generate precise uniform or specified sampling [Surazhsky and Gotsman 2003], but it fails to provide an easy way to globally sample the mesh in accordance with a density function. Other works placed the initial point set by a mesh adaption process [Yue et al. 2007] or by error diffusion algorithm on mesh surfaces [Alliez et al. 2003]. Afterwards, Lloyd relaxation is applied on local overlapping parameterizations [Surazhsky et al. 2003] or on a global parameter domain [Alliez et al. 2003] to improve the initial point distribution. The Lloyd-based algorithm can generate precise isotropic distribution in accordance with a density function. However, the initial vertex sampling is delicate to control and needs to performed with care. Recently, Schreiner et al.[2006] proposed an algorithm based on advancing-front paradigm to perform local remeshing directly on 3D surface or locally to a region of interest, which can produce high-quality meshes.

3 Preliminaries

The input of our remeshing scheme is assumed to be orientable two-manifolds of arbitrary genus. In this paper, we mainly focus on closed meshes, while it is trivial to handle boundary case by a small modification of our scheme. The input mesh is presumed to be a piecewise linear approximation of smooth surface, but with unsatisfactory mesh quality.



(a) (b)

Figure 1: *Feature edges detection of two models*

For meshes with sharp features, such as CAD models, the features must be preserved after remeshing. Before sampling, we should extract a set of feature edges and corners from the input mesh first. Since the input mesh is a discrete approximation of the underlying smooth surface, feature extraction may be sensitive to noise. We adopt the method proposed by Jiao and Heath [2002] which can perform efficient and reliable feature detection. Strong edges on a discrete mesh are first identified by the dihedral angles of edges and artificial feature edges are removed by checking the geometry of neighborhood. As shown in Figure 1, skeleton for CAD model as well as ridges and corners for smooth models can be retrieved from discrete meshes. Both feature edges and corners require special treatment during the subsequent processing.

4 Poisson-Disk Sample on 3D Meshes

Poisson disk distribution is regarded as one of the best sampling patterns for its blue noise property [Dippé and Wold 1985; Cook 1986; Lagae and Dutré 2006b]. Many approaches have been developed to generate Poisson disk distribution [McCool and Fiume 1992; Lagae and Dutré 2006a; Kopf et al. 2006; White et al. 2007]. The fast Poisson-disk sample generation algorithm proposed by Dunbar and Humphreys [2006] is efficient to directly generate maximal Poisson-disk distribution with excellent blue noise characteristic. In this paper, we aim to extend this fast sampling algorithm to two-manifolds in 3D so that it can generate isotropic point distribution on mesh surface. Before describing our algorithm, we will give a short review on the fast 2D Poisson-disk sampling algorithm first.

4.1 Fast Poisson-disk Sample Generation in 2D Domain

Dart-throwing [Cook 1986] is one of the simplest but slowest techniques to generate Poisson-disk distribution. It iteratively refines an existing point set by generating random point locations in the sample domain. A point is discarded if there already exists another point within a disk with certain radius r of it. This approach is slow because it wastes a lot of time to generate points that might be discarded. Moreover, the generated point set is usually not maximal, i.e., some region may be undersampled. To alleviate this problem and make the approach more efficient, Dunbar and Humphreys [2006] developed a modified dart throwing algorithm – maximized boundary sampling. It runs in $O(n)$ time and is guaranteed to terminate. The essential idea of their method is to allow sampling only the regions where it is legal to place a dart. The sub-domain within which it is legal to add a new point is called *available domain*. Let $D(\mathbf{p}, r)$ be a disk of radius r around a point $\mathbf{p}$. We name the disk as *expellant disk*. Then, the available domain for a domain X and a point set P is:

$$S_X = X - \bigcup_{\mathbf{p} \in P} D(\mathbf{p}, 2r)$$

Based on the observation that it is not necessary to sample the entire available domain, Dunbar and Humphreys further restricted the available region for a single point into an *available boundary* $\mathcal{B}(\mathbf{p}, r)$ – the boundary of expellant disk. Then the available boundary of a new point $\mathbf{p}$ for P can be defined as:

$$\mathcal{B}_{\mathbf{p}} = \mathcal{B}(\mathbf{p}, 2r) - \bigcup_{\mathbf{p}' \in P} \mathcal{B}(\mathbf{p}', 2r)$$

The entire available boundary is $\mathcal{B} = \bigcup_{\mathbf{p} \in P} \mathcal{B}_{\mathbf{p}}$, which is composed of a set of circular arcs. Figure 2 illustrates the entire available boundary after three points are inserted.

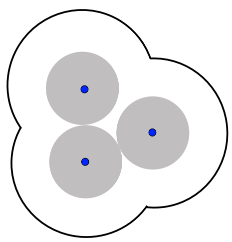


Figure 2: *Available boundary of three points*

4.2 Uniform Poisson-disk Sample over 3D Mesh Surface

To extend the fast Poisson-disk sampling algorithm to 3D surface, an associated distance metric should be well defined. Because the models to be processed may contain various details, it is not ideal to simply use Euclidean distance in $\mathcal{R}^3$ to compute distance over the surface. Euclidean distance approximates geodesic distance, namely the distance metric between the points along the surface, only in regions of low curvature. Otherwise, they are of great difference. In order to achieve precise isotropic sampling on surface, we adopt geodesic distance as the distance metric. Thus the available boundary of a sampling point on 3D surface will be a geodesic equidistant curve.

Suppose the input mesh $\mathcal{M}$ is closed, the Poisson-disk sampling algorithm over 3D mesh can be proceeded as the following steps:

1. Randomly choose a point $\mathbf{p}_0$ on the surface as the first sample point;
2. Compute the equidistant curve $\mathcal{B}_0$ of $\mathbf{p_0}$ on $\mathcal{M}$ using the method described in section 4.2.2. The radius of the equidistant circle is $2r$, where r is the radius of expellant disk and can be deduced from the formula given in section 4.2.1. $\mathcal{B}_0$ will act as the initial set of available boundary;
3. Generate a random point $\mathbf{p}_i$ on the available boundary [Anderson 1993];
4. Compute the equidistant curve of $\mathbf{p}_i$ and update the available boundary as described in section 4.2.3;
5. Repeat steps 3 and 4 until the set of the entire available boundary is empty;

Our approach progressively inserts new vertices onto the surface. After termination, the distribution on $\mathcal{M}$ will be uniform in terms of geodesic distance. Adaptive sampling in accordance with density will be discussed in section 5.

4.2.1 Relationship Between Density and Disk Radius

Suppose that N is the number of points over a unit domain on $\mathcal{M}$. The absolute radius r of the expellant disk around each vertex is defined as [Lagae and Dutré 2006b]:

$$r_{max} = \sqrt{1/(2\sqrt{3}N)}, \quad r = \rho \cdot r_{max} \tag{1}$$

where r_{max} is the maximum possible radius. The density is assumed to be linearly proportional to the density in the case of compact packing by hexagonal lattice and ρ is usually a constant for a specific algorithm. The assumption is validated by our experiments in uniform sampling case. The value of ρ will also be deduced in the experiments as shown in section 7.

4.2.2 Computing Geodesic Equidistant Curve

We use the algorithm proposed by Surazhsky et al. [2005] to compute the geodesics on meshes. It can compute geodesic distance accurately and efficiently. The kernel of the algorithm is a "window propagation" scheme. A *window* is actually an interval on the mesh edge. Each window is associated with a source $\mathbf{v}_s$ (also a pseudosource $\mathbf{s}$ for saddle points). The geodesic paths in the same window pass the same sequence of triangle stripes and can be unfolded into the same plane. Thus, the distance field $D(\mathbf{p})$ over a window $w : (b_0, b_1)$ is explicitly defined as:

$$D(\mathbf{p}) = \|\mathbf{p} - \mathbf{s}'\| + \sigma \tag{2}$$

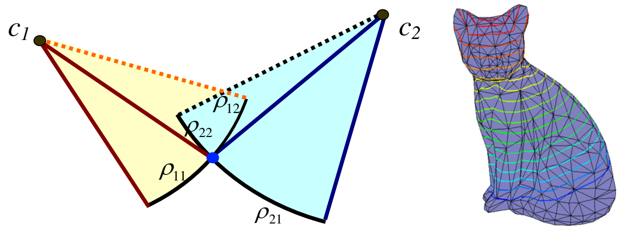
where $\mathbf{s}'$ is the unfolding of $\mathbf{v}_s$ or $\mathbf{s}$ on the plane w lies in, σ is the geodesic distance between $\mathbf{v}_s$ and $\mathbf{s}$. In our application, $\mathbf{v}_s$ is the newly sampled point.

To compute the distance from a source point to all the vertices on a mesh, the windows propagate in a wavefront way implemented by a priority queue. Since we only need to compute the equidistant circle, it is not necessary to perform window propagation until the whole mesh is covered by windows. If the minimum distance of w to $\mathbf{v}_s$ is greater than $2r$, then the window is not pushed into the priority queue for further propagation. This strategy makes the propagation terminate faster.

Equidistant circular arcs only exist in those windows whose maximum distance is no less than $2r$. For this kind of windows, a circle with radius of $2r$ is intersected with the triangle formed by the endpoints of the window and its unfolded source to extract the intersection circular arcs. The union set $\mathcal{C}$ may contain point set whose geodesic distance to the source is less than $2r$, because the distance field for points that are interior of the same triangle may overlap, see Figure 3(a) for example. To eliminate these artificial equidistant circular arcs, the arcs in the same triangle are intersected with each other, and the arc portions that lie in the sector of other arcs are trimmed. The remaining arcs will form a closed path that constitutes the equidistant curve. Figure 3(b) shows multi-equidistant curves with increasing distance on a cat model.

4.2.3 Updating Available Boundary

Assume that the circular arcs in the available boundary are arranged in clockwise order from end to end. After the equidistant curve $\mathcal{B}_{new}$ of a new sample point is extracted, the whole available boundary $\mathcal{B}_{whole}$ should be updated accordingly. $\mathcal{B}_{whole}$ may contain some separate closed curves.



(a) (b)

Figure 3: *(a) ρ_{12} and ρ_{22} should be scissored because they lie in the sector of other arcs. (b) Some equidistant curves on a cat model.*

To update $\mathcal{B}_{whole}$, we first compute the intersection points of $\mathcal{B}_{new}$ and $\mathcal{B}_{whole}$. Since each arc that constitutes the available boundary is situated within only one triangle, the intersection computation can be localized by performing the intersection only for those arcs situated in the same triangle. After the intersection, $\mathcal{B}_{whole}$ and $\mathcal{B}_{new}$ are divided into different parts at the intersection points. In this way, the parts that are to be kept and to be discarded are arranged alternatively. That is, if an arc part is to be kept/discarded, then its adjacent part must be discarded/kept.

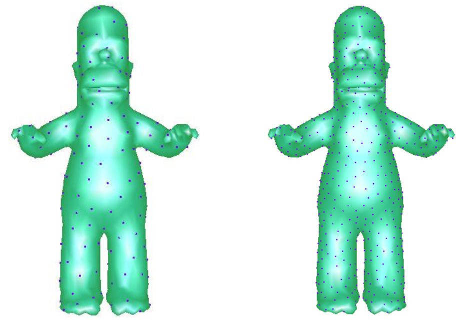
For each arc, we record the index of the triangle it lies in. The indices set of the triangles that $\mathcal{B}_{whole}$ covers is denoted as $\mathcal{T}_{whole}$. Similarly, the indices set of $\mathcal{B}_{new}$ is recorded as $\mathcal{T}_{new}$. To accelerate the update process, we adopt the following strategies to determine which part of circular arcs is to be discarded or to be kept for each closed curve in $\mathcal{B}_{whole}$:

- If there is one arc B_k whose triangle element T_k satisfies: $T_k \in \mathcal{T}_{whole}$ but $T_k \notin \mathcal{T}_{new}$, then the arc must be outside the covered region of $\mathcal{B}_{new}$, therefore B_k must be kept.
- If $\mathcal{T}_{whole} \subseteq \mathcal{T}_{new}$, we find in $\mathcal{B}_{whole}$ the arc B_k that the new sample lies in. Since B_k must lie in the interior region of $\mathcal{B}_{new}$, B_k must be discarded.
- If both of the above two strategies fail, we get the midpoint $\mathbf{q}$ of the longest arc B_k in $\mathcal{B}_{whole}$. Then the region that $\mathcal{B}_{new}$ covers is conformally parameterized to 2D plane [Desbrun et al. 2002]. $\mathcal{B}_{new}$ and $\mathbf{q}$ are also mapped into the parameter domain as $\mathcal{B}'_{new}$ and $\mathbf{q}'$. We count the number N of intersection points between $\mathcal{B}'_{new}$ and a line passing $\mathbf{q}'$. If N is odd, then B_k lies in the region covered by $\mathcal{B}_{new}$ and must be discarded. Otherwise, B_k is kept.

After determining the property of B_k, the property of the arc part containing it has the same property with it. Afterwards, all the arc parts in $\mathcal{B}_{whole}$ can be determined to be kept or discarded by assigning different properties to adjacent parts.

To determine the properties of arcs in $\mathcal{B}_{new}$, we first find the location of an intersection point $\mathbf{q}_0$ in $\mathcal{B}_{new}$. Since all the arcs in the available boundary are arranged in clockwise order, we can determine the property of the arc B'_k which begins with $\mathbf{q}_0$ by the role of $\mathbf{q}_0$ in the kept part of $\mathcal{B}_{whole}$: If $\mathbf{q}_0$ is a start point in the kept part of $\mathcal{B}_{whole}$, then B'_k should be discarded, or it will be kept. Afterwards, the properties of the rest arcs in $\mathcal{B}_{new}$ can also be determined in the same way as those in $\mathcal{B}_{whole}$.

Figure 4 illustrates the results of uniform Poisson-disk sampling with different densities over a 3D mesh surface. Since the radius of expellant disk in Figure 4(a) is double of that in Figure 4(b), the number of samples in Figure 4(b) is four times as many as that in Figure 4(a) as expected.



(a) 212 samples (b) 847 samples

Figure 4: *The model is uniformly sampled with different densities. The radius of expellant disk in (a) is double of that in (b).*

4.2.4 Feature Preservation

If there are feature edges and corners on the surface, they should be preserved in the result. Dense sampling on features not only significantly increases the number of sampling points but also causes aliasing problems. In order to preserve the features, we perform 1-dimensional feature sampling before random sampling. Firstly, the feature corners are sampled and their available boundaries are computed. Then, we distribute sampling points on the feature edges with uniform intervals and compute the available boundary of each point. Finally, the available boundaries of feature corner samples and feature edge samples are united to act as the initial available boundary for random sampling (See Figure 5 for example).

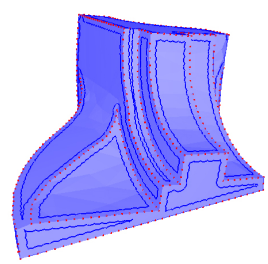


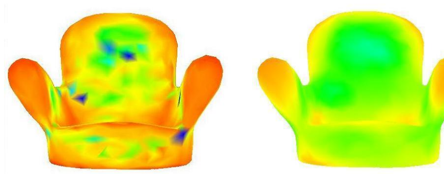
Figure 5: *The set of closed curves is the available boundary after sampling the features on the Fandisk model, and the red dots are the sampling points on the features.*

5 Adaptive Poisson-Disk Sampling

The algorithm presented in section 4 samples the input mesh with expellant disk of constant radius, which results in uniformly distributed sampling points. The overall density of the sampling point set is inversely proportional to r which can be adjusted to tune the density of the point set. Therefore, our approach can be extended to adaptive sampling. Since the radii of points in different regions on surface are different, the available boundary for a point is no longer a geodesic equidistant curve.

5.1 Available Boundary of Adaptive Sampling

Assume that the density function d over $\mathcal{M}$ is specified by user or deduced from geometric quantity (e.g. curvature) measured on the given discrete model. In our experiments, we use differential geometry techniques [Rusinkiewicz 2004] to estimate curvature on each vertex. The user can control the final mesh gradation by specify a gamma function and the parameters of a low-pass filter over the density function. Since the curvature is approximated by discrete techniques, it may be sensitive to both geometry and connectivity of $\mathcal{M}$, especially for those meshes with sparse vertices. Hence a Laplacian smoothing is applied on d to achieve smooth gradation (Figure 6).



(a) (b)

Figure 6: *Density map before(a) and after(b) Laplacian smoothing*

Till now, the density $d(\mathbf{p}_i)$ on each vertex $\mathbf{p}_i$ is well-defined, accordingly the radius $r(\mathbf{p}_i)$ of its corresponding expellant disk can be deduced by Eq.1. Also the radius of an arbitrary point on surface can be calculated by linear interpolation. Suppose $T_i = (\mathbf{p}_{j_1}, \mathbf{p}_{j_2}, \mathbf{p}_{j_3})$ is a triangle on $\mathcal{M}$. (All the coordinates used in this section are expressed in the local 2D coordinate of the plane where T_i lies.) The corresponding coordinates of $\mathbf{p}_{j_k}$ are $(x_k, y_k), k = 1, 2, 3$. Let $\mathbf{p} = (x, y)$ be a point inside T_i, then the expellant disk radius of $\mathbf{p}$ can be written as:

$$r(\mathbf{p}) = r(x, y) = ax + by + c' \tag{3}$$

It can be deduced that a, b, c' are determined linearly by $r(\mathbf{p}_k)$ and the coordinates of $\mathbf{p}_{j_k}, k = 1, 2, 3$.

Suppose the sample point is $\mathbf{p}_0$. The available boundary over 3D surface is essentially defined to be the set of points $\mathbf{p}$ whose distance to $\mathbf{p}_0$ is the sum of the expellant disk radii at $\mathbf{p}$ and $\mathbf{p_0}$. When combined with Eq. 2 and Eq. 3, the available boundary of $\mathbf{p}_0(x_0, y_0)$ is expressed as follows:

$$\|\mathbf{p} - \mathbf{p_0}\| + \sigma = r(\mathbf{p}_0) + ax + by + c' \tag{4}$$

let $c = r(\mathbf{p}_0) + c' - \sigma$, then the above equation of the available boundary can be written as:

$$(x - x_0)^2 + (y - y_0)^2 = (ax + by + c)^2 \tag{5}$$

That is, the available boundary is the x-y plane projection of the intersection between a cone $z = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ and a plane $z = ax + by + c$, which means the projection is a conic curve. Therefore, the available boundary of adaptive sampling points is composed of conic arcs.

5.2 Finding the Available Boundary

The approach of finding available boundary is similar to the algorithm described in section 4.2.2, but a small modification is needed, since the available boundary is no longer an equidistant curve. In the uniform sampling case, the termination condition of window propagation is judged by the minimum geodesic distance of the window. While in the adaptive sampling case, we judge the termination by intersecting the conic of the source $\mathbf{s}$ with the triangle formed by $\mathbf{s}$ and the endpoints $(\mathbf{q}_1, \mathbf{q}_2)$ of the window.

If intersection exists, yet none lies on the window, we do not propagate the window further. If there is at least one intersection point lying on the window, the window is put into the priority queue for further propagation. In the situation that there is no intersection, we test if the geodesic distance from $\mathbf{q}_1$(or $\mathbf{q}_2$) to $\mathbf{s}$ is larger than $r(\mathbf{s}) + r(\mathbf{q}_1)$(or $r(\mathbf{s}) + r(\mathbf{q}_2)$). If the judgement is true, the propagation terminates, otherwise, it continues. This modified window propagation is also performed until the priority queue is empty.

6 Sampling Points Relocation

Precise isotropic sampling can be achieved by Lloyd relaxation [LLOYD 1983], which has been proved quite successful in 2D sampling, e.g. [McCool and Fiume 1992; Ostromoukhov 2007], because Lloyd relaxation minimizes an energy related to the compactness of Voronoi cells and distributes the energy equally in each cell [Gersho 1979]. Weighted centroidal Voronoi diagrams [Du et al. 2006] even allows to define a density function for each Voronoi cell.

As in the 2D sampling algorithm, we will also apply a few passes of Lloyd relaxation to the point set to improve the isotropy. Before Lloyd relaxation, the connectivity of the sampling points should first be recovered. There are many mesh reconstruction algorithm proposed for point sets [Hoppe et al. 1992; Cheng et al. 2004], but they cannot guarantee the topology unchanged, especially for those tubular shapes, such as tails and legs of animals, see Figure 7 for example. In this paper, we will reconstruct the mesh by mutual tessellation [Turk 1992], which makes use of the topology information of the input mesh and can guarantee the topology of remeshed surface.

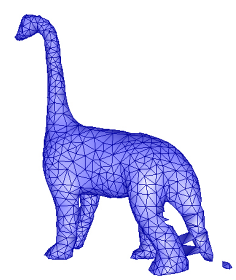


Figure 7: *Topology is not guaranteed after reconstructing from the sampling point set.*

6.1 Mesh Reconstruction From Sampling Point Set

To restore the connectivity, we incorporate both the sampling points and the original vertices on the input surface. First, all the sampling points are inserted into the mesh by splitting the triangles they lie in, which may produce a lot of skinny triangles. To make the algorithm more stable, we perform a run of edge flip on the new mesh $\mathcal{I}$ to improve the triangle quality. An edge is a candidate flipping edge if the flip will increase the minimal angle of the triangles sharing the edge. To guarantee the validity of topology, the edge can be flipped if the new edge after flipping does not exist in the original mesh and the change of normal direction and dihedral angle cannot exceed an angle of β and γ respectively. The value of β and γ can be specified by the user. If the above requirements are not met, topology errors might occur.

In the second step, the original vertices are removed from $\mathcal{I}$. Suppose $\mathbf{v}$ is the vertex to be removed and $\mathbf{v}_i, i = 1, \cdots, m$ are the 1-ring neighboring vertices of $\mathbf{v}$. We parameterize the neighborhood to 2D domain in the following way [Floater 1997]: Assume $\mathbf{p}, \mathbf{p}_1, \cdots, \mathbf{p}_m$ are the projections of $\mathbf{v}, \mathbf{v}_1, \cdots, \mathbf{v}_m$ respectively. We first set $\mathbf{p} = \mathbf{0}$ and $\mathbf{p}_1 = (\|\mathbf{v}_1 - \mathbf{v}\|, 0)$, then compute

$\mathbf{p}_i, i = 2, \cdots, m$ sequentially to satisfy:

$$\|\mathbf{p}_i - \mathbf{p}\| = \|\mathbf{v}_i - \mathbf{v}\|,$$
$$\mathbf{ang}(\mathbf{p}_i, \mathbf{p}, \mathbf{p}_1) = \frac{\sum_{j=1}^{i-1} \mathbf{ang}(\mathbf{v_j}, \mathbf{v}, \mathbf{v_{j+1}})}{\sum_{j=1}^{m} \mathbf{ang}(\mathbf{v_j}, \mathbf{v}, \mathbf{v_{j+1}})} \cdot 2\pi$$

The parameterized sub-mesh $\mathcal{S}'(\mathbf{v})$ can well preserve the original shape of triangles. After removing $\mathbf{p}$, the left part is tessellated by constrained Delaunay triangulation [Shewchuk 1996] with restrictions that no new edges are generated out of $\mathcal{S}'(\mathbf{v})$ and the boundary edges of $\mathcal{S}'(\mathbf{v})$ must be included in the final triangulation. With these restrictions, even concave polygon can be handled correctly. Then the triangulation is projected back to $\mathcal{I}$.

To guarantee the topology fidelity of the new mesh, extra check should be made during removing to avoid that two sides of the surface are accidentally joined together, which often occurs in a tubular region formed by a few polygons. For example in Figure 8, if vertex V is removed and edge CE is generated in the re-triangulation, the mesh will be changed into a non-manifold one. We check this problem by examining whether the newly generated edge already exists in the k-ring neighborhood of V. k is set to 3 in our implementation. If the problem occurs, the vertex is not removed. The mesh obtained after all original vertices are removed is denoted as $\mathcal{M}'$. Figure 9(d) shows an example $\mathcal{M}'$ of a Maxplanck model.

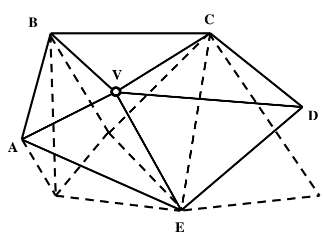


Figure 8: *Topology may be changed after V is removed (See section 6.1 for detail). The solid lines show 1-ring neighborhood of V. The dotted mesh is a part of surface below V.*

6.2 Vertex Location optimization by Lloyd Relaxation

To optimize the location of the sampling points in our algorithm, Lloyd relaxation is performed locally to move each non-feature sampling point $\mathbf{v}_i$ in $\mathcal{M}'$ to the center of the centroids of the adjacent faces of $\mathbf{v}_i$, i.e. the weighted center of Voronoi Tessellation. This strategy is similar to [Alliez et al. 2003; Surazhsky et al. 2003; Yue et al. 2007] in spirit.

Instead of work explicitly on the Voronoi diagram, the relaxation is performed on the dual triangulation. First, the 1-ring neighborhood of $\mathbf{v}$ is parameterized into 2D domain in the same way as we have used in section 6.1. Then the Voronoi region of $\mathbf{p}$ is constructed in the parameter domain and $\mathbf{p}$ is moved to :

$$\mathbf{p}' = \frac{\sum_{j=1}^{m} d(j) \cdot \mathbf{c}_j}{\sum_{j=1}^{m} d(j)} \quad (6)$$

where $\mathbf{c}_j$ is the centroid of the j-th triangle T_{i_j} incident on $\mathbf{v}_i$, and $d(j)$ is the density of $\mathbf{c}_j$, which can be linearly interpolated in T_{i_j}. If the sub-mesh is concave, the new position of $\mathbf{p}'$should be guaranteed to lie in the sub-mesh with some perturbations. Afterwards, $\mathbf{p}'$ is projected back to the input mesh surface using the method proposed in [Surazhsky et al. 2003] to prevent shape derivations. Then $\mathbf{v}_i$ is moved to the new location. To maintain the local Delaunay property, edge flip in 3D is performed after several steps of Lloyd relaxation.

After a run of Lloyd relaxation, the isotropy of the sampling points will be improved effectively. Usually about 3 runs of relaxation are enough to generate weighted centroidal Vornoi tessellation, thus producing precise isotropic sampling. By redistributing the sampling points, the regularity of triangle faces is also improved (See Figure 9(e)).

6.3 Mesh Connectivity Optimization

Edge flip operation is always used to improve the regularity of mesh connectivity [Alliez et al. 2002; Yue et al. 2007]. A non-feature edge is flipped if the flip reduces the following energy function:

$$\mathcal{E}(\mathcal{M}) = \sum_{v \in \mathcal{M}} (d(v) - o(v))^2 \quad (7)$$

where $d(v)$ is the degree of v and $o(v)$ is the ideal degree of v. Most researchers set $o(v)$ to 4 for boundary vertices and 6 for interior ones. In our work, to maximize the minimal angle and to improve the regularity of triangle shape, we set $o(v)$ for each interior vertex as:

$$o(v) = \frac{\sum_{i \in N_v} \mathbf{ang}(v_i v v_{i+1})}{60}, \quad (8)$$

to make each angle approximate its ideal angle of 60°, where N_v is the 1-ring neighborhood of v. We then alternate between the optimizations of geometry and connectivity to improve the mesh quality.

7 Experimental Results

We have implemented the algorithm proposed in this paper on a Pentium IV PC(2.8GHz) with 512 RAM. The input mesh can be remeshed uniformly or adaptively with respect to surface curvature. In addition, the density for adaptive sampling can be adjusted by several parameters and can be smoothed to achieve smooth mesh gradation.

Value of ρ To determine the value of ρ in Eq.1, we first let $\rho = 1$. For $N = 500 \sim 5000$ per unit with a step of 100, we compute the corresponding radii by Eq.1 and then sample a unit square domain with our algorithm. The numbers N' of actual sampling points are recorded and then compared with N as shown in Figure 10(a). It can be seen that N' is approximately linearly proportional to N. According to the statistics, the average of ratio N'/N is 0.7124 with covariance of 2.2936×10^{-5}. So we adjust ρ to 0.8440(the square of 0.7124) and recompute the radii. The actual numbers of sampling points are well coincident with the ideal ones. Figure 10 (b) shows the curve of Eq.1 and the experimental relationship between radii and actual number of sampling points. These two curves fit well. Therefore, we can approximately estimate the expellant radius for sampling from Eq.1 with $\rho = 0.8440$.

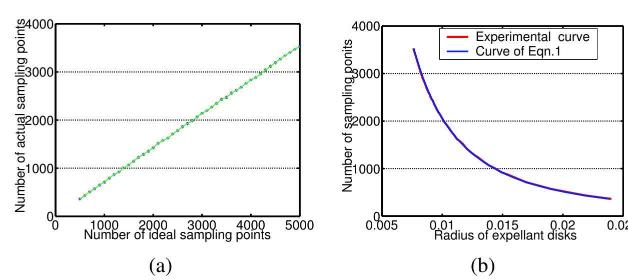


Figure 10: *Estimation of ρ from experiments.*

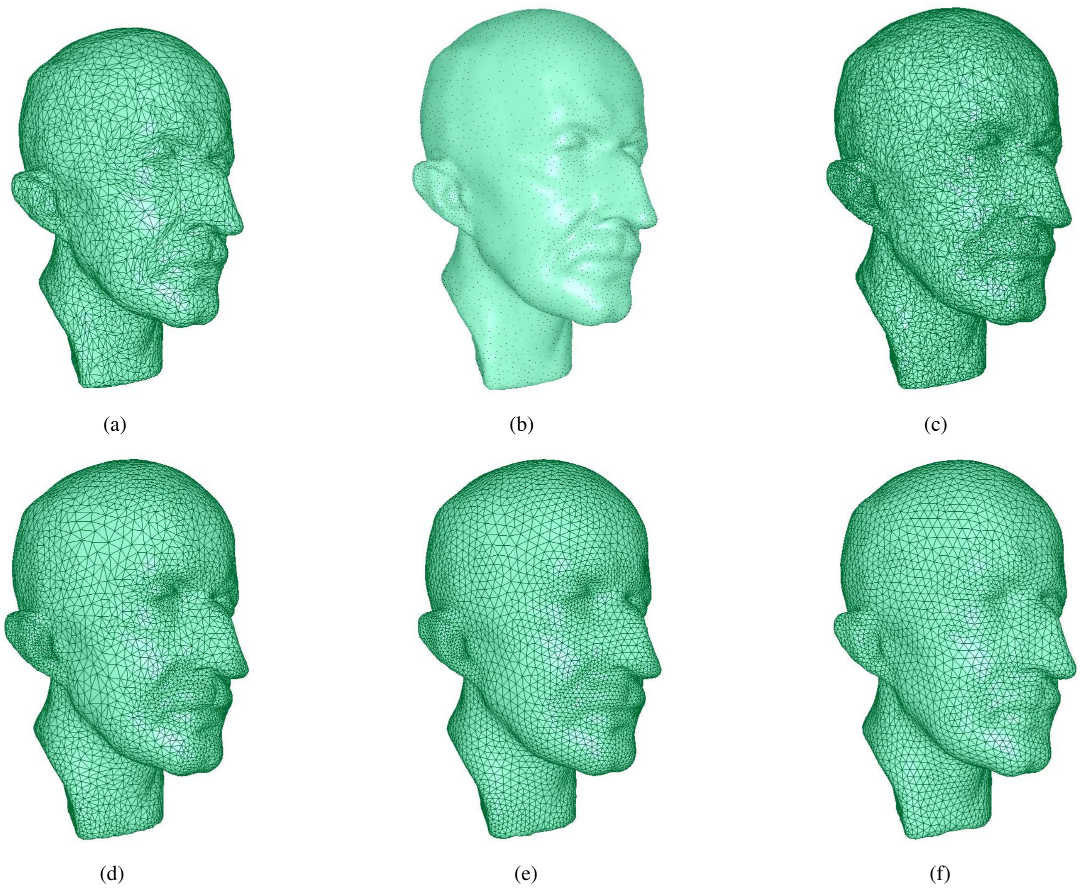


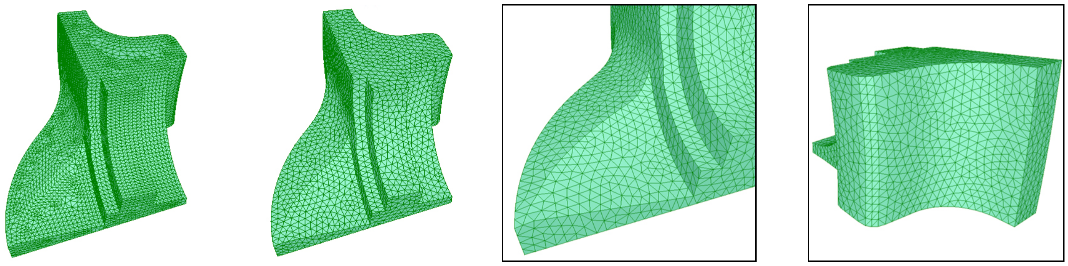
Figure 9: *Remeshing of Maxplanck model.(a) original model. (b) curvature-adapted sampling points. (c) mutual tessellation result. (d) mesh after removing original vertices. (e) mesh after Lloyd relaxation of (d). (f) mesh optimized using the method in [Surazhsky et al. 2003].*

Examples Figure 9 shows the remeshing process of the Maxplanck model. The original surface is sampled with curvature-adapted density, as illustrated in Figure 9(b). After reconstructing from the sampling points (Figure 9(d)), the mesh is optimized by Lloyd relaxation and edge-flip technique into a high-quality new mesh (Figure 9(e)). Figure 11 illustrates the uniform remeshing of a fandisk model with 6475 vertices. The creases and corners of the model are well-preserved in the new model with 3121 vertices.

Statistics We have also applied our algorithm to various models with arbitrary genus and complexity. Some of them come from marching cubes process, and others are results of simplification process. The remeshing results are illustrated in Figure 12. To evaluate the quality of the results, we collected the statistics of the minimal angles of the resulting triangles. The minimum and the average of the minimal angles are reported in the second row of each item in Table 1. For most models, we achieve a minimum angle above 30° and the average angles are mostly above 50°, which indicate the high quality of the resulting meshes. The quality of the whole mesh is illustrated with histograms of the angle distribution of all triangles. Since our work focuses on the improvement of triangle shape, the connectivity regularity is not comparable to some remeshing approaches, but it can be greatly improved by further up-to-date connectivity optimizations. The executing time for sampling and remeshing is also listed in the table. We believe the time cost is acceptable for remeshing as a preprocess for most applications though the code has not been optimized. Since each step of our algorithm ensures that the samples lie on the original surface, the resulting mesh is expected to have a high fidelity to the original mesh, which is validated by METRO [Cignoni et al. 1998], a tool used to measure the error between two models. Results show most remeshing errors are within 10^{-3}.

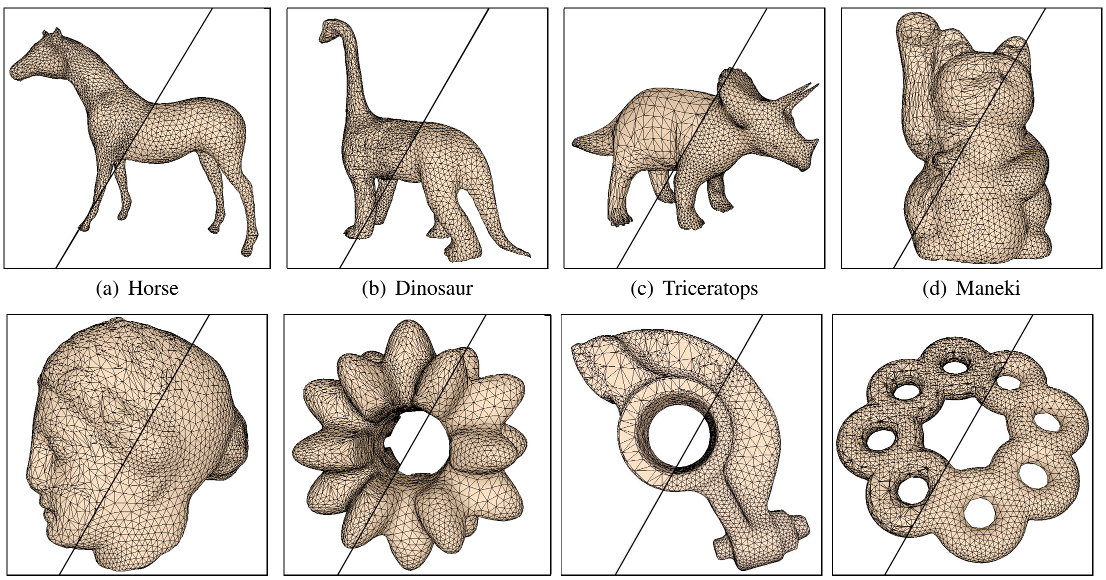
Comparison We also compare our remeshing algorithm with the one proposed in [Surazhsky et al. 2003]. With the same density definition, we used the vertices on the original surface as the initial sampling and then solely applied Lloyd relaxation on it. Since the initial sampling process is omitted, the implementation of algorithm in [Surazhsky et al. 2003] is much more efficient than ours. Most of the running time of our algorithm is spent on the initial vertex placement. The time of placement optimization in both algorithms is nearly same. The result mesh of Maxplanck model is demonstrated in Figure 9 (f). It is shown that our sampling algorithm can control the placement of vertices better. Since the samples are generated according to the density map over the mesh surface, our algorithm cannot provide exact control on the number of generated samples. The statistics of regularity of shape and connectivity are shown in the third row of each item in Table 1, which shows our algorithm can provide better results.

Discussion of stability It is know that the computation of geodesic distances on low quality meshes might lead to numerical instabilities. In our implementation, we also find that the geodesic isoline of a sample might be not closed when the radius is too large, which will cause the failure of the union of available boundary. The fail rate is around 0.3% on average in our experiments. In this



(a) (b) (c) (d)

Figure 11: *Uniform remeshing of Fandisk model. (a)Original model (b) the remeshed model. Details are shown in (c-d).*



(a) Horse (b) Dinosaur (c) Triceratops (d) Maneki

(e) Hygiea (f) BumpyTorus (g) Rocker-arm (h) Nine-Torus

Figure 12: *Comparison between models before and after remeshing*

case, we simply discard the sample. Since the sample is selected randomly, this would not cause any problem but make our algorithm more robust. The random sequence [Anderson 1993] used in our implementation will generate a new sample with less probability that the next sample is close to the previous one geometrically, which reduces the probability of failure of the new sample.

8 Conclusion and Future Work

We have proposed a novel technique to generate isotropic sampling on a two-manifold triangle mesh. By successfully extending the fast Poisson-Disk sampling algorithm to 3D mesh surface, we can generate uniform or adaptive distributions directly on 3D manifold surface. Thanks to the isotropy of point distribution generated by our sampling algorithm, only a few runs of relaxation are needed to optimize the precise location of the sampling points, which not only guarantees the quality of triangle shape but also makes the procedure more efficient. The regularity of geometry and connectivity is also improved in the process of Lloyd iterations combined with edge-flip technique. As illustrated in the experimental results, the result meshes after remeshing are of higher quality compared with Lloyd-standalone techniques, although we paid higher computation time. Since most of the time cost is spent on the computation of the accurate geodesic isolines, it can be reduced by approximately compute the geodesic isolines as the available boundary.

The samples are centered in the expellant disks, therefore the proposed sampling technique can also be applied to perform object distribution on 3D mesh surfaces. Moreover, progressive multi-level sampling is easy to achieve by gradually reducing the radius and recomputing the available boundary after a coarser level sampling is finished. In this way, all the vertices in the lower-detailed models will be present in the models with more details, which can be used to smoothly interpolate between the different levels of detail.

As future work, we plan to investigate the way to generate anisotropic sampling points on 3D surface. In anisotropic case, the expellant region will not be a disk but an elliptical region. How to define the ellipse according to the directions of principle curvature is remained to be explored.

Acknowledgements

Thanks to the anonymous reviewers who have made helpful suggestions for improving this paper. Some of the original models are from the AIM@SHAPE Repository.

This work was supported in part by National Science Foundation of China Grant No. 60573149 and National Science Foundation of Beijing Grant No. 4072013.

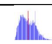
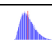
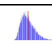
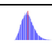
Model	Vertex number	Minimum∠	Average∠	Angle histogram	Irregular rate	Time cost(s)
Maxplanck(original)	6143	3.577°	36.29°		0.61	
Maxplanck(our)	7601	32.33°	52.17°		0.260	222
Maxplanck([Surazhsky et al. 2003])	6143	23.53°	48.32°		0.29	8.4
Horse(original)	5064	6.75°	39.47°		0.415	
Horse(our)	3017	35.70°	51.91°		0.274	103
Horse([Surazhsky et al. 2003])	5064	19.83°	46.97°		0.337	5.5
Dinosaur(original)	5903	1.68°	34.21°		0.597	
Dinosaur(our)	3365	31.01°	51.87°		0.276	68
Dinosaur([Surazhsky et al. 2003])	5903	26.92°	49.46°		0.313	4.4
Triceratops(original)	2832	0.002°	29.57°		0.593	
Triceratops(our)	6529	27.54°	51.62°		0.285	133
Triceratops([Surazhsky et al. 2003])	2823	15.67°	47.13°		0.316	6.2
Maneki(original)	5027	0.784°	33.79°		0.607	
Maneki(our)	4139	31.32°	52.00°		0.253	112
Maneki([Surazhsky et al. 2003])	5027	24.54°	47.29°		0.302	6.7
Hygiea(original)	4500	1.265°	34.94°		0.65	
Hygiea(our)	3092	35.43°	51.99°		0.257	113
Hygiea([Surazhsky et al. 2003])	4500	29.18°	50.10°		0.308	4.3
BumpyTorus(original)	10417	0.47°	35.92°		0.497	
BumpyTorus(our)	4236	31.74°	51.76°		0.287	416
BumpyTorus([Surazhsky et al. 2003])	10417	14.77°	47.33°		0.282	16.1
Rocker-arm(original)	3431	0.042°	34.07°		0.638	
Rocker-arm(our)	3551	31.63°	51.53°		0.275	106
Rocker-arm([Surazhsky et al. 2003])	3343	18.40°	48.43°		0.354	5.6
Nine-Torus(original)	4688	1.55°	32.5°		0.69	
Nine-Torus(our)	4040	32.55°	51.86°		0.284	128
Nine-Torus([Surazhsky et al. 2003])	4688	24.67°	48.52°		0.314	3.8

Table 1: *Statistics of resulting meshes. The histograms are shown with the same proportion of width and the red bars in the histogram show the position of* 60°

References

ALLIEZ, P., MEYER, M., AND DESBRUN, M. 2002. Interactive geometry remeshing. *ACM Transactions on Graphics, 21*, 3, 347–354.

ALLIEZ, P., ÉRIC COLIN DE VERDIÈRE, DEVILLERS, O., AND ISENBURG, M. 2003. Isotropic surface remeshing. In *SMI '03: Proceedings of the Shape Modeling International 2003*, IEEE Computer Society, Washington, DC, USA, 49–58.

ALLIEZ, P., UCELLI, G., GOTSMAN, C., AND ATTENE, M. 2005. Recent advances in remeshing of surfaces,. Research report, AIM@SHAPE Network of Excellence.

ANDERSON, P. 1993. Linear pixel shuffling for image processing: an introduction. *Journal of Electronic Imaging, 2*, 2, 147–154.

BOISSONNAT, J. D., AND OUDOT, S. 2003. Provably good surface sampling and approximation. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 9–18.

BOTSCH, M., PAULY, M., ROSSL, C., BISCHOFF, S., AND KOBBELT, L. 2006. Geometric modeling based on triangle meshes. In *ACM SIGGRAPH 2006 Courses*, ACM Press, New York, NY, USA, 1.

CHENG, S.-W., DEY, T. K., RAMOS, E. A., AND RAY, T. 2004. Sampling and meshing a surface with guaranteed topology and geometry. In *SCG '04: Proceedings of the twentieth annual symposium on Computational geometry*, ACM Press, New York, NY, USA, 280–289.

CIGNONI, P., ROCCHINI, C., AND SCOPIGNO, R. 1998. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum, 17*, 2, 167–174.

COOK, R. L. 1986. Stochastic sampling in computer graphics. *ACM Transactions on Graphics, 5*, 1, 51–72.

DESBRUN, M., MEYER, M., AND ALLIEZ, P. 2002. Intrinsic parameterizations of surface meshes. *Computer Graphics Forum, 21*, 3, 209–218.

DIPPÉ, M. A. Z., AND WOLD, E. H. 1985. Antialiasing through stochastic sampling. *Computer Graphics (Proceedings of ACM SIGGRAPH 1985), 19*, 3, 69–78.

DU, Q., EMELIANENKO, M., AND JU, L. 2006. Convergence of the lloyd algorithm for computing centroidal voronoi tessellations. *SIAM Journal on Numerical Analysis, 44*, 1, 102–119.

DUNBAR, D., AND HUMPHREYS, G. 2006. A spatial data structure for fast poisson-disk sample generation. *ACM Transactions on Graphics, 25*, 3, 503–508.

FLOATER, M. S., AND HORMANN, K. 2005. Surface parameterization: a tutorial and survey. In *Advances in multiresolution for geometric modelling*, N. A. Dodgson, M. S. Floater, and M. A. Sabin, Eds. Springer Verlag, 157–186.

FLOATER, M. S. 1997. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design, 14*, 4, 231–250.

GERSHO, A. 1979. Asymptotically optimal block quantization. *IEEE Transactions on Information Theory, 25*, 4 (Jul), 373–380.

GU, X., GORTLER, S. J., AND HOPPE, H. 2002. Geometry images. *ACM Transactions on Graphics, 21*, 3, 355–361.

HOPPE, H., DEROSE, T., DUCHAMP, T., MCDONALD, J., AND STUETZLE, W. 1992. Surface reconstruction from unorganized points. *Computer Graphics (Proceedings of ACM SIGGRAPH 92), 26*, 2, 71–78.

JIAO, X., AND HEATH, M. T. 2002. Feature detection for surface meshes. In *Proceedings of 8th International Conference on Numerical Grid Generation in Computational Field Simulations*, 705–714.

JONES, T. R. 2006. Efficient generation of poisson-disk sampling patterns. *Journal of Graphics Tools, 11*, 2, 27–36.

KOPF, J., COHEN-OR, D., DEUSSEN, O., AND LISCHINSKI, D. 2006. Recursive wang tiles for real-time blue noise. *ACM Transactions on Graphics, 25*, 3, 509–518.

LAGAE, A., AND DUTRÉ, P. 2006. An alternative for wang tiles: colored edges versus colored corners. *ACM Transactions on Graphics, 25*, 4, 1442–1459.

LAGAE, A., AND DUTRÉ, P. 2006. A comparison of methods for generating Poisson disk distributions. Report CW 459, Department of Computer Science, K.U.Leuven, Leuven, Belgium, August.

LEE, A. W. F., SWELDENS, W., SCHRÖDER, P., COWSAR, L., AND DOBKIN, D. 1998. Maps: multiresolution adaptive parameterization of surfaces. In *In Proceedings of ACM SIGGRAPH 1998*, ACM Press / ACM SIGGRAPH, New York, NY, USA, Computer Graphics Proceedings, Annual Conference Series, 95–104.

LLOYD, S. 1983. An optimization approach to relaxation labeling algorithm. *Image and Vision Computing, 1*, 2, 85–91.

MCCOOL, M., AND FIUME, E. 1992. Hierarchical poisson disk sampling distributions. In *Proceedings of the conference on Graphics interface '92*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 94–105.

OSTROMOUKHOV, V. 2007. Sampling with polyominoes. *ACM Transactions on Graphics, 26*, 3, 78–83.

PEYRÉ, G., AND COHEN, L. D. 2006. Geodesic remeshing using front propagation. *International Journal of Computer Vision, 69*, 1, 145–156.

RUSINKIEWICZ, S. 2004. Estimating curvatures and their derivatives on triangle meshes. In *Proceedings of the 3D Data Processing, Visualization, and Transmission*, IEEE Computer Society, Washington, DC, USA, 486–493.

SCHREINER, J., SCHEIDEGGER, C. E., FLEISHMAN, S., AND SILVA, C. T. 2006. Direct (re)meshing for efficient surface processing. *Computer Graphics Forum, 25*, 3, 527–536.

SHEWCHUK, J. R. 1996. Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. In *FCRC '96/WACG '96: Selected papers from the Workshop on Applied Computational Geormetry, Towards Geometric Engineering*, 203–222.

SURAZHSKY, V., AND GOTSMAN, C. 2003. Explicit surface remeshing. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 20–30.

SURAZHSKY, V., ALLIEZ, P., AND GOTSMAN, C. 2003. Isotropic remeshing of surfaces: a local parameterization approach. Tech. Rep. 4967, INRIA. ECG-TR-304100-01.

SURAZHSKY, V., SURAZHSKY, T., KIRSANOV, D., GORTLER, S. J., AND HOPPE, H. 2005. Fast exact and approximate geodesics on meshes. *ACM Transactions on Graphics, 24*, 3, 553–560.

TURK, G. 1992. Re-tiling polygonal surfaces. *Computer Graphics (Proceedings of ACM SIGGRAPH 92), 26*, 2, 55–64.

VORSATZ, J., RÖSSL, C., AND SEIDEL, H.-P. 2003. Dynamic remeshing and applications. In *ACM Symposium on Solid Modeling and Applications*, ACM Press, 167–175.

WHITE, K. B., CLINE, D., AND EGBERT, P. K. 2007. Poisson disk point sets by hierarchical dart throwing. In *IEEE Symposium on Interactive Ray Tracing*, 129–132.

YU, Y., ZHOU, K., XU, D., SHI, X., BAO, H., GUO, B., AND SHUM, H.-Y. 2004. Mesh editing with poisson-based gradient field manipulation. *ACM Transactions on Graphics, 23*, 3, 644–651.

YUE, W., GUO, Q., ZHANG, J., AND WANG, G. 2007. 3d triangular mesh optimization in geometry processing for cad. In *SPM '07: Proceedings of the 2007 ACM symposium on Solid and Physical Modeling*, ACM Press, New York, NY, USA, 23–33.

ZHOU, B., AND FANG, X. 2003. Improving mid-tone quality of variable-coefficient error diffusion using threshold modulation. *ACM Transactions on Graphics, 22*, 3, 437–444.