Spectral sampling of manifolds
Extended version

ETH Zurich, Department of Computer Science, Technical Report No. 683, September 2010. All rights reserved.

Spectral Sampling of Manifolds: Extended Version

Cengiz Öztireli* ETH Zürich
Marc Alexa† TU Berlin
Markus Gross‡ ETH Zürich

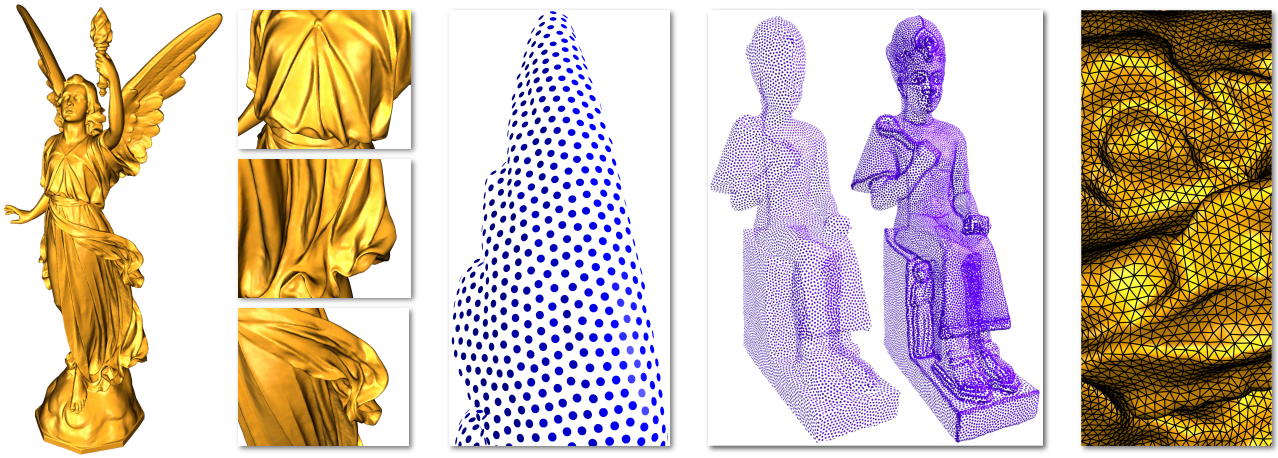

Figure 1: *From left to right: The original model with 14 million samples is adaptively subsampled to 300k samples and reconstructed, total time including out-of-core simplification and reconstruction is 6 minutes. Detail from uniform resampling of the model to 21k samples. Uniform and adaptive samplings. Remeshing obtained by using sampled points as vertices.*

Abstract

A central problem in computer graphics is finding optimal sampling conditions for a given surface representation. We propose a new method to solve this problem based on spectral analysis of manifolds which results in faithful reconstructions and high quality isotropic samplings, is efficient, out-of-core, feature sensitive, intuitive to control and simple to implement. We approach the problem in a novel way by utilizing results from spectral analysis, kernel methods, and matrix perturbation theory. Change in a manifold due to a single point is quantified by a local measure that limits the change in the Laplace-Beltrami spectrum of the manifold. Hence, we do not need to explicitly compute the spectrum or any global quantity, which makes our algorithms very efficient. Although our main focus is on sampling surfaces, the analysis and algorithms are general and can be applied for simplifying and resampling point clouds lying near a manifold of arbitrary dimension.

CR Categories: G.1.2 [Numerical Analysis]: Approximation—Approximation of surfaces and contours; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations;

Keywords: Sampling, Laplace-Beltrami, Heat Kernel, Poisson Disk Sampling

1 Introduction

Approximating or interpolating a manifold or a function from scattered data is an essential problem for reconstruction or smoothing purposes. One of the fundamental issues for scattered data approximation is finding an optimal sampling that ensures accurate reconstructions with minimum redundancy. The sampling problem is especially hard for general manifolds since no parametric domain is defined and standard methods from signal processing do not apply [Saucan et al. 2008].

There have been significant efforts in spectral graph theory and computational harmonic analysis domains to extend signal processing results to discrete approximations of manifolds by utilizing techniques from spectral analysis [Lafon and Lee 2006]. Following the seminal work of Taubin [1995], these developments have been inspiring many successful algorithms in geometry processing [Zhang et al. 2007; Sun et al. 2009]. Despite their excellent performance and theoretical foundations, spectral methods are global and need eigendecomposition of a large matrix. Although there have been local patching approches to approximate the spectral properties [Pauly and Gross 2001], finding a decomposition of a manifold into patches from which *global spectral properties* can be derived is a difficult problem. Global nature of spectral methods makes their application to massive datasets unfeasible even though specialized methods are employed [Vallet and Lévy 2008]. This has been hindering use of current spectral approaches in a sampling algorithm.

In this paper, we overcome this drawback and propose sampling algorithms that utilize new measures derived from spectral analysis of manifolds. Although our measures conceptually measure *global* changes to the manifold, they only require *local* computations. This allows us to use the theoretical framework within efficient algorithms. Since we do not assume any knowledge about connectivity between the point samples, our methods can be applied directly without costly meshing operations.

Combining results from spectral analysis, kernel methods and ma-

*e-mail: cengizo@inf.ethz.ch
†e-mail: marc.alexa@tu-berlin.de
‡e-mail: grossm@inf.ethz.ch

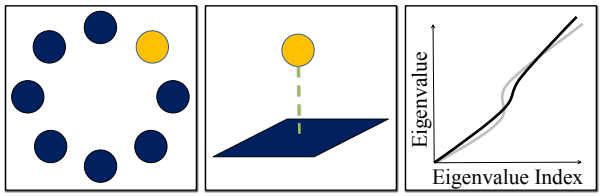

Figure 2: *Conceptial Overview: (a) Given a set of points, (b) we map them to a higher dimensional space implied by the heat kernel of the underlying manifold such that the distance of a point to the span of others can be used to measure the (c) influence of that point on the Laplace-Beltrami spectrum.*

trix perturbation theory, we derive our local measure that quantifies the change a point causes to the manifold, if it is added to the point set defining the manifold. The derived measure has strong connections to approximation by kernel regression. Hence, generated samplings result in accurate reconstructions using kernel based algorithms such as Moving Least Squares (MLS) based Point Set Surfaces (PSS) [Levin 2003; Alexa et al. 2003]. If our initial simplification method is followed by our resampling scheme derived from the same measure, distribution of point samples also possesses high quality blue noise properties, which are desirable in many applications such as remeshing [Fu and Zhou 2008] and anti aliasing [Cook 1986]. Although our main focus in this paper is on low dimensional manifolds, our algorithms are general such that they can be applied to manifolds of arbitrary dimension.

To summarize, our main contributions are the following:

- Out-of-core simplification and in-core resampling algorithms for point sampled manifolds that are efficient, simple to implement, easy to control through intuitive parameters, feature sensitive, result in accurate reconstructions with kernel based approximation methods and high quality isotropic samplings.
- A *discrete spectral analysis* of manifolds using results from kernel methods and matrix perturbation theory.

After reviewing prior approaches to the problem (Section 2), we give an overview of our approach (Section 3), derive our measure (Section 4) and show its connection to kernel based approximation (Section 5). We then develop sampling algorithms based on the measure (Section 6) and illustrate the properties of the resulting samplings and reconstructions (Section 7). Finally, we conclude the paper with a discussion of the approach and possible future directions for research (Section 8).

2 Related Work

Sampling in Computer Graphics

Efficient simplification of data is essential to process large datasets. The fastest methods for simplification are based on clustering of points [Pauly et al. 2002]. They are simple to implement and out-of-core, thus suitable for very large datasets. However, sampling quality is not sufficient for accurate reconstructions. An efficient and still accurate set of sampling algorithms simplify the point set by iteratively removing or adding a sample at a time based on a measure derived from the geometry and sampling rate. This measure can be defined as the distance of a point to the surface [Alexa et al. 2001; Carr et al. 2001], density and curvature based heuristics [Ohtake et al. 2004; Kitago and Gopi 2006] or quadratic error metrics [Garland and Heckbert 1997; Pauly et al. 2002]. The algorithms provide better accuracy for the final surfaces, but computing the initial scores and updating can become prohibitively expensive [Kitago and Gopi 2006].

After simplifying the data, further resampling can be applied to improve quality. A common approach to generate high quality isotropic samplings of a surface is distributing points using relaxation techniques. These methods first compute an initial distribution of points on the surface, and then refine this distribution by techniques such as variants of the Lloyd's method [Yan et al. 2009; Valette et al. 2008; Alliez et al. 2003], particle systems [Turk 1992; Witkin and Heckbert 1994], or advancing front algorithms [Schreiner et al. 2006]. In spite of their good sampling properties, computational cost or critical dependence on parameters or initial distributions hinder their use. Instead of operating on the surface, some relaxation algorithms first parametrize the surface and then use well established methods [Lagae and Dutré 2008] to generate distributions with blue noise characteristics on the parameter domain [Alliez et al. 2002]. However, parametrization is a hard problem that may introduce distortions. This has led to algorithms that directly compute distributions with blue noise properties on a meshed surface using geodesic distances [Fu and Zhou 2008], albeit at a high computational cost.

Another class of resampling algorithms greedily place new samples so as to satisfy some constraints. Farthest point sampling [Boissonnat and Oudot 2003] places a new sample at a time such that it is maximally away from all other inserted points. For meshes, the algorithm has been extended using geodesic distances to improve both sampling quality and running time [Peyré and Cohen 2006]. However, these greedy algorithms generally perform worse than relaxation based methods while computational cost remains high.

Sampling for Manifold Learning

Apart from the focus on 2-manifolds in computer graphics, sampling problem has also been tackled for general higher dimensional manifolds and ambient spaces. The term manifold learning refers to discovering the intrinsic manifold structure of a given point set [Coifman and Lafon 2006]. Many learning techniques first construct a weighted graph from the points and form the graph Laplacian matrix. Eigendecomposition of this matrix can then be used to estimate embeddings of the data points into a new space such that various learning tasks can be done easily [Ham et al. 2004]. Since these operations are computationally very demanding, several methods for sampling datasets have been proposed, but so far they mostly depend on random sampling [Karoui and D'Aspremont 2009; Drineas and Mahoney 2005; Achlioptas et al. 2001]. Notable exceptions try to exploit the low-rank structure of the diffusion matrix [Coifman and Lafon 2006] or find subsets of the samples and use the Nyström method to approximate the eigenvectors [Liu et al. 2006]. However, the algorithms are computationally expensive and not suitable for high quality samplings of manifolds.

3 Overview

Before going into the details of the derivation of our measure, we would like to summarize the essential ingredients of our approach from a practical point of view. The theoretical ideas and details can be found in Sections 4 and 5. The reader who is more interested in the algorithms can skip those sections and jump to Section 6 after reading this section.

The input to our sampling method is a set of points lying near a manifold with normals and a kernel function definition (Figure 3 (a)). If the normals are not provided, they can be estimated by fitting local proxy surfaces [Guennebaud and Gross 2007].

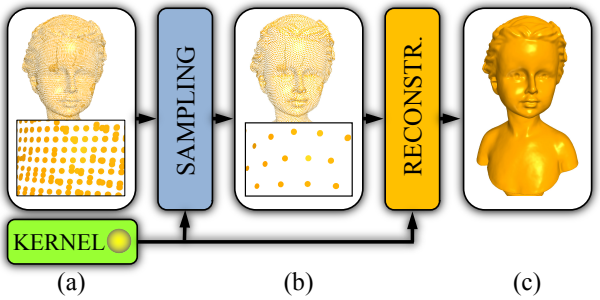

Figure 3: *(a) Input to our sampling method is a set of points with normals and a kernel function. (b) Sampling algorithms are run on the data set based on the spectral measure computed using the kernel function. (c) Resulting samplings can be used to accurately reconstruct surfaces using the same kernel.*

The data together with the kernel function can be used to define a continuous surface. We chose to use the adaptive kernel function definition of the robust and feature sensitive variant of an MLS based PSS [Öztireli et al. 2009]. Let $\mathbf{x} = [\mathbf{p}/\sigma_p \ \mathbf{n}/\sigma_n]^T$, where $\mathbf{p}$ is the position of a point, $\mathbf{n}$ is the normal vector at that point and σ_p and σ_n are user provided smoothness parameters, then the kernel used for sampling and reconstruction is given by $k(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2}$. Higher values of σ_p leads to smoother surfaces and lower values of σ_n causes more pronounced sharp features. This kernel definition allows us to generate adaptive samplings without having to set individual shapes or bandwidths for the kernels by operating on a 2-dimensional manifold embedded in $\mathbb{R}^6$ [Lai et al. 2007].

Once the kernel is fixed, our simplification and resampling algorithms can be run on the input data (Figure 3 (b)). For both algorithms (Section 6), we utilize our measure derived from the spectral analysis of manifolds as explained in Section 4. The resulting distribution of points has high quality blue noise properties and the sampled data can be used to accurately reconstruct a continuous surface with the PSS [Öztireli et al. 2009] (Figure 3 (c)). The same kernel with the same parameters σ_p and σ_n is used for both sampling and reconstruction.

4 Measuring the importance of a sample

The essence of our method is measuring the effect of a point on the manifold using the Laplace-Beltrami spectrum. It is well-known that eigenvalues of the Laplace-Beltrami operator provide an almost unique identification of the manifold up to isometry [Reuter et al. 2006; Rustamov 2007]. Although there exist isospectral manifolds (i.e. sharing the same spectrum) that are not isometric, these cases are very rare, and many geometric and topological properties of a manifold can be extracted from the spectrum [Kesavan 1998; Lévy 2006]. The eigenspectrum is also stable under perturbations of the manifold, thus similar manifolds will have close spectra [Dey et al. 2010] and changes in the implicit structure of the manifold will be reflected to the spectrum. We utilize these facts to measure changes in a manifold through changes in its spectrum. Measuring this change due to a single point is performed by considering a higher dimensional space implied by the heat kernel (Figure 2).

In this section, we will show how to compute the effect of adding a point x to the point set defining the surface on the Laplace-Beltrami spectrum efficiently. First we present our choice of the discrete approximation for the spectrum of the Laplace-Beltrami operator via the heat kernel. Next, we derive our measure in terms of the heat kernel, and then show how to compute it.

4.1 Laplace-Beltrami and its discretization based on the heat operator

The Laplace-Beltrami operator is a generalization of the Laplace operator in $\mathbb{R}^d$ to Riemannian manifolds. For our purposes, we will concentrate on compact manifolds without boundary. The eigenfunctions $u_i(x)$ and eigenvalues λ_i of this operator are the solutions of

$$\triangle u_i(x) = \lambda_i u_i(x), \tag{1}$$

where $x \in M$ for a manifold M and $\triangle$ denotes the Laplace-Beltrami operator (on M). Eigenfunctions form a complete orthonormal basis for functions on the manifold and can be used for harmonic analysis similar to the Fourier basis in $\mathbb{R}^d$, and eigenvalues are analogous to frequencies.

In order to compute the Laplace-Beltrami operator in practice, we need to discretize it. This problem has been intensively studied, resulting in a variety of solutions. For samples forming a simplicial mesh, several discrete approximations with nice invariance and convergence properties have been derived [Pinkall and Polthier 1993; Cohen-Steiner and Morvan 2006; Wardetzky et al. 2007; Bobenko and Springborn 2007; Bauer et al. 2009]. However, during sampling or re-sampling we wish to avoid repeatedly constructing such a mesh, as it can be a challenging problem in itself. We rather follow the line of approximations motivated by methods in manifold learning, spectral graph theory, and computational harmonic analysis [Belkin and Niyogi 2006; Coifman and Lafon 2006]. These approaches rely on the connection between the Laplace-Beltrami operator and the heat operator.

The heat operator H_t gives the heat distribution $H_t f(x)$ at time t for an initial distribution $f(x)$ on the manifold. The relation between H_t and $\triangle$ is

$$H_t = e^{-t\triangle} \tag{2}$$

It can be shown that for small t the exponential operator indeed converges to its first order terms [Belkin and Niyogi 2006], i.e.

$$\triangle = \lim_{t \to 0} \frac{I - H_t}{t} \tag{3}$$

Thus for short time, eigenvalues of H_t can be used to approximate those of $\triangle$. For point clouds and graphs, the estimates of $\triangle$ depend on approximations of H_t and thus equation (3) is critical for these approximations to converge [Belkin and Niyogi 2006].

Associated with the heat operator, there is the heat kernel $h_t(x, y) : M \times M \times \mathbb{R}^+ \to \mathbb{R}^+$ that satisfies

$$H_t f(x) = \int_M h_t(x, y) f(y) dy \tag{4}$$

where $f : M \to \mathbb{R}$ and dy is the volume form on M. A familiar heat kernel is that of the Euclidean space $\mathbb{R}^d$ given by the Gaussian $1/(4\pi t)^{d/2} e^{-\|\mathbf{x}-\mathbf{y}\|/4t}$ (Note, however, that the kernel for an arbitrary manifold M is not readily available). Hence, intuitively the operator H_t smoothes functions on the manifold by convolving them with the heat kernel.

In order to compute an approximation of the heat operator, we can discretize equation (4) to get

$$(\mathbf{H}_t \mathbf{f})_i = \sum_j h_t(x_i, x_j) \mathbf{f}_j \tag{5}$$

where $\mathbf{f}_j = f(x_j)$. The entries of this heat kernel matrix are thus given by $(\mathbf{H}_t)_{ij} = h_t(x_i, x_j)$. By expanding the heat kernel using eigenfunctions and eigenvalues of the Laplace-Beltrami operator

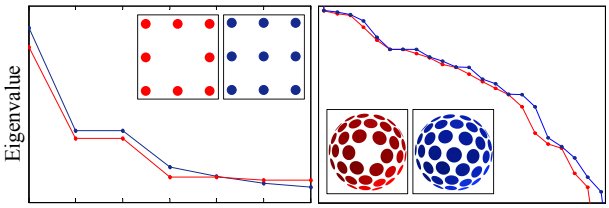

Figure 4: *Effect of adding a point to the point set on the eigenvalues of the heat kernel matrix for (a) $\mathbb{R}^2$ and (b) sphere (zoomed).*

(e.g. [Grigor'yan 1998]) it can be shown [Braun 2006; von Luxburg 2004] that the eigenvalues of the matrix $\mathbf{H}_t$ converge to those of H_t up to a constant, assuming the eigenvectors of $\mathbf{H}_t$ are good approximations to those of H_t, which is the case as the sample set becomes denser.

Concluding, the eigenvalues of the Laplace-Beltrami operator can be approximated by computing those of the discrete heat operator, i.e. the matrix $\mathbf{H}_t$ as $t \to 0$. It remains to measure the change in the spectrum of $\mathbf{H}_t$.

4.2 Effect of a Point on the Spectrum

Given points x_i on the manifold that are used to form $\mathbf{H}_t$, we seek to quantify the change in the spectrum of $\mathbf{H}_t$ due to addition of a new point x. We reformulate this problem in another space and use matrix perturbation theory and properties of the eigenfunctions of the heat kernel to arrive at our measure.

In machine learning, it is common to use expansions of symmetric positive semi-definite kernels to define a possibly infinite dimensional feature space where dot products are given by the kernel [Schölkopf et al. 1998]. The expansion of the heat kernel (e.g. [Grigor'yan 1998]) allows us to use a similar approach. By rewriting the expansion we get

$$h_t(x,y) = \sum_{i=0}^{\infty} e^{-\lambda_i t} u_i(x) u_i(y) = \phi_t(x)^T \phi_t(y) \tag{6}$$

Here, $\phi_t(x)$ is a vector such that the i^{th} component is given by $\sqrt{e^{-\lambda_i t}} u_i(x)$. This interpretation allows us to define a new space where these vectors live. For brevity, we will drop the time dependency from all identities and write $\phi_i = \phi_t(x_i)$ for the rest of this section. We also denote eigenvalues of a matrix $\mathbf{M}$ as $\lambda_i(\mathbf{M})$.

With these definitions, the heat kernel matrix can be written as $\mathbf{H}_{ij} = h(x_i, x_j) = \phi_i^T \phi_j$. More interestingly, we can define a covariance matrix $\mathbf{C} = \sum \phi_i \phi_i^T$. Non-zero eigenvalues $\lambda_i(\mathbf{C})$ of $\mathbf{C}$ are the same as those of $\mathbf{H}$ [Schölkopf et al. 1998]. Thus we can equivalently consider the change in the spectrum of $\mathbf{C}$.

Adding a sample x to the point set means forming a new covariance matrix $\mathbf{C}' = \mathbf{C} + \phi\phi^T$ with $\phi = \phi(x)$. The vector ϕ can be written as $\phi = \mathbf{r} + \mathbf{o}$ for the projection $\mathbf{r}$ onto the span of ϕ_i's and the orthogonal component $\mathbf{o}$. Thus we can expand the expression for $\mathbf{C}'$ as $\mathbf{C}' = \mathbf{C} + (\mathbf{r}+\mathbf{o})(\mathbf{r}+\mathbf{o})^T = \mathbf{C} + \mathbf{o}\mathbf{o}^T + \mathbf{E}$. The matrix $\mathbf{C} + \mathbf{o}\mathbf{o}^T$ has eigenvalues λ_i and $\|\mathbf{o}\|^2$. We want to study how the eigenspectrums of $\mathbf{C} + \mathbf{o}\mathbf{o}^T$ and $\mathbf{C}'$ differ.

Let $\mathbf{v}_i$ denote an eigenvector of $\mathbf{C}$ corresponding to the eigenvalue $\lambda_i(\mathbf{C})$. The change in the eigenvalues can be studied using the following result from matrix perturbation theory [Ipsen and Nadler 2009]:

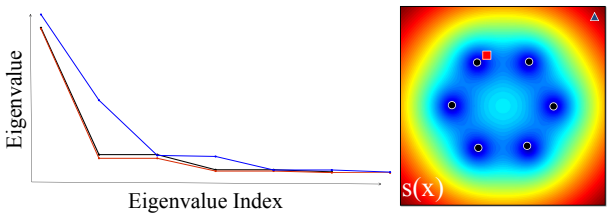

Figure 5: *On the left, black curve shows the spectrum of the heat kernel matrix and on the right, the measure $s(x)$ is plotted for the six black points and the heat kernel of $\mathbb{R}^2$ (blue:low, red:high). When a new point (represented by the blue triangle) in a region of high $s(x)$ is added to the set of black points, eigenspectrum becomes the blue curve, and hence changes significantly. In contrast, another point in a region of low $s(x)$ (red square) results in the red curve on the left and hence does not cause much change.*

$$\min_j |\lambda_i(\mathbf{C} + \mathbf{o}\mathbf{o}^T) - \lambda_j(\mathbf{C}')|^2 \leq \|\mathbf{E}\mathbf{v}_i\|^2 \tag{7}$$

Substituting the expression for $\mathbf{E}$, the bound can be computed as

$$\|\mathbf{E}\mathbf{v}_i\|^2 = \|(\phi\mathbf{r}^T + \mathbf{r}\mathbf{o}^T)\mathbf{v}_i\|^2 = \|\phi\|^2(\mathbf{r}^T\mathbf{v}_i)^2$$

where we used the fact that $\mathbf{o}$ is orthogonal to $\mathbf{v}_i$. Thus the change in $\lambda_i(\mathbf{C})$ due to adding the point x to the set depends on $\delta_i = (\mathbf{r}^T\mathbf{v}_i)^2 = (\phi^T\mathbf{v}_i)^2$. In Apppendix A, we derive that if the eigenvectors of $\mathbf{H}$ well approximate eigenfunctions of the heat operator, the change in the spectrum will diffuse to many eigenvalues and effect of a point on the spectrum will not be significant. Examples of this behavior are shown in Figure 4 for the cases of $\mathbb{R}^2$ and sphere.

This analysis implies that if $\sum \delta_i = \|\mathbf{r}\|^2$ is close to $\|\phi\|^2$, x will not disturb the distribution of the eigenvalues considerably. On the other hand, if $\|\mathbf{o}\|^2$ is large, then the spectrum will have a new large eigenvalue, which will alter the distribution of the eigenvalues. Thus we define our measure as $\|\mathbf{o}\|^2/\|\phi\|^2 \in [0,1]$. Although one could define other measures based on the individual bounds δ_i, this particular choice results in very efficient computations as will be shown in Section 6. In Appendix B, we show that this measure can be computed as

$$s(x) = 1 - \mathbf{h}^T\mathbf{H}^{-1}\mathbf{h}/h(x,x) \tag{8}$$

where $(\mathbf{h})_i = h(x, x_i)$. We illustrate that $s(x)$ correctly captures changes to the spectrum in Figure 5. If $s(x)$ is low at a location, placing a new sample at that location does not change the spectrum much.

Note that since we do not know the heat kernel of a given manifold, a direct computation of this expression is not possible. In the next section, we utilize approximation methods from spectral processing of graphs to arrive at our final measure.

4.3 Computing the Measure

For a given set of points $\{\mathbf{x}_i \in \mathbb{R}^d\}_1^n$ lying near a manifold, graph based methods try to approximate the Laplace-Beltrami operator of the manifold with that of the weighted graph constructed from the point set. It has been shown that under the assumption that the points are sampled according to a uniform probability distribution on the manifold, this approximation converges to the Laplace-Beltrami operator as the sample set becomes denser [Belkin and Niyogi 2006; Coifman and Lafon 2006]. One of the fundamental

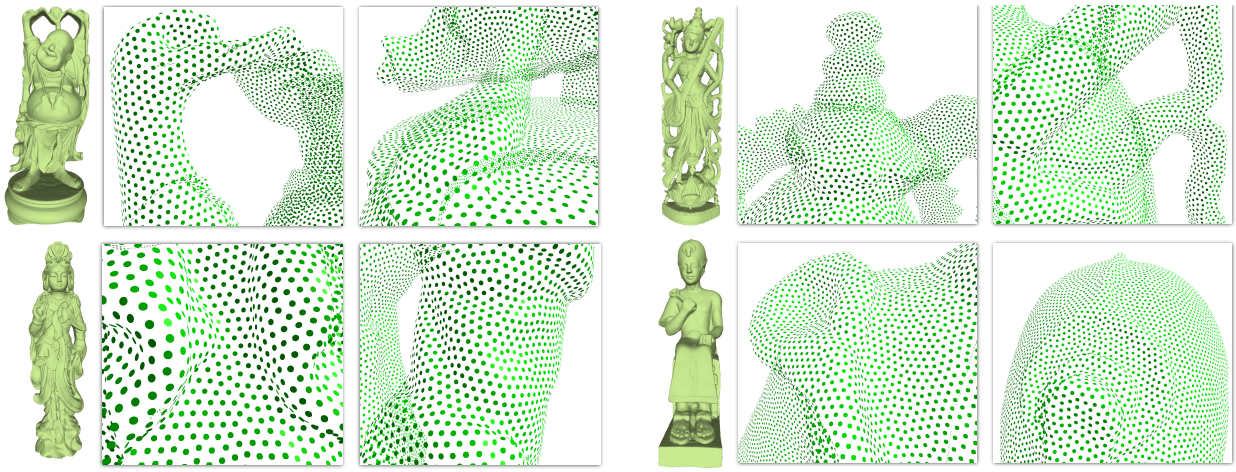

Figure 6: *Uniform samplings of various models by our algorithm.*

observations in their proof is that heat kernel operator of the manifold is well-approximated by the matrix $\mathbf{D}^{-1}\mathbf{K}$, for the Gaussian kernel $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$ encoding similarity between points, the kernel matrix $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and the diagonal matrix $\mathbf{D}$ with entries $\mathbf{D}_{ii} = \sum_{j=1}^{n} \mathbf{K}_{ij}$. Hence, spectrum of the matrix $\mathbf{D}^{-1}\mathbf{K}$ provides an approximation for that of the heat kernel matrix $\mathbf{H}$.

It is easy to show that the matrices $\mathbf{D}^{-1}\mathbf{K}$ and $\mathbf{D}^{-1/2}\mathbf{K}\mathbf{D}^{-1/2}$ share the same eigenvalues. Since we are only interested in the eigenvalues, we can equivalently use the latter in our analysis. Let us call the latter matrix as $\tilde{\mathbf{H}}$.

Although we could directly substitute $\tilde{\mathbf{H}}$ for $\mathbf{H}$ in the definition of $s(x)$, it is not clear how to define $(\mathbf{h})_i = h(x, x_i)$. Instead, we consider the continuous kernel from which the entries of $\tilde{\mathbf{H}}$ can be generated by sampling and discretization [von Luxburg 2004; Coifman and Lafon 2006]

$$\tilde{h}(\mathbf{x}, \mathbf{y}) = k(\mathbf{x}, \mathbf{y}) / \sqrt{d(\mathbf{x}) d(\mathbf{y})}$$

where $d(\mathbf{x}) = \int k(\mathbf{x}, \mathbf{y}) \mu(\mathbf{y})$ for some measure μ. Let us define the diagonal matrix $(\bar{\mathbf{D}})_{ii} = d(\mathbf{x}_i)$ and the vector $\mathbf{k} = [k(\mathbf{x}, \mathbf{x}_1) \cdots k(\mathbf{x}, \mathbf{x}_n)]^T$. By direct substitution of $\tilde{h}$ for h in the expression for $s(x)$ (equation (8)), we can compute the following expression

$$\tilde{s}(\mathbf{x}) = 1 - \left(\frac{1}{\sqrt{d(\mathbf{x})}} \mathbf{k}^T \bar{\mathbf{D}}^{-1/2} \bar{\mathbf{D}}^{1/2} \mathbf{K}^{-1} \bar{\mathbf{D}}^{1/2} \bar{\mathbf{D}}^{-1/2} \mathbf{k} \frac{1}{\sqrt{d(\mathbf{x})}} \right) / \frac{k(\mathbf{x}, \mathbf{x})}{d(\mathbf{x})}$$

which leads to the final expression for our measure

$$\tilde{s}(\mathbf{x}) = 1 - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k} / k(\mathbf{x}, \mathbf{x}) \tag{9}$$

In a sampling algorithm, computing $\tilde{s}(\mathbf{x})$ will require inversion of the global matrix $\mathbf{K}$ many times as more points are added or removed, which will make the algorithm very inefficient. Fortunately, this global matrix can be substituted by a local matrix constructed from the neighbors of $\mathbf{x}$. The Gaussian kernel k is a positive definite and symmetric kernel. Hence, by Mercer's theorem, k can be written as a dot product $k(\mathbf{x}, \mathbf{y}) = \varphi(\mathbf{x})^T \varphi(\mathbf{y})$ [Schölkopf et al. 1998]. As derived in Appendix B, $\mathbf{k}^T \mathbf{K}^{-1} \mathbf{k}$ is the norm of the projection of $\varphi(\mathbf{x})$ onto the span of $\varphi(\mathbf{x}_i)$'s. The Gaussian k is approximately locally supported, meaning that $k(\mathbf{x}, \mathbf{x}_i) = \varphi(\mathbf{x})^T \varphi(\mathbf{x}_i) \approx 0$ for $\|\mathbf{x} - \mathbf{x}_i\| > r$ for a support radius r. This implies that the vectors $\varphi(\mathbf{x}_i)$ for points $\|\mathbf{x} - \mathbf{x}_i\| > r$ will be almost orthogonal to $\varphi(\mathbf{x})$. Hence, the projection of $\varphi(\mathbf{x})$ onto the span of all vectors can be well-captured by the projection onto the span of local neighbors of $\mathbf{x}$. Thus $\tilde{s}(\mathbf{x})$ can be well approximated by using $\mathbf{K}$ and $\mathbf{k}$ constructed from the neighboring points. An example of this behaviour is shown in Figure 8. Since number of neighboring points in the support of k for a point $\mathbf{x}$ will be much smaller than total number of points in the set, this local computation makes our sampling very efficient.

5 Relation to Kernel Regression

For many problems in computer graphics, sampling is followed by reconstruction or inferring new data. Reconstructions from sampled data are desired to match that of the full data. A versatile approach to surface reconstruction is using weighted averages of kernels. Prominent examples are Radial Basis Function and MLS based reconstruction methods. In particular, we are using a feature preserving implicit MLS surface [Öztireli et al. 2009]. Since our measure also involves a kernel based approximation, it is closely connected to these methods.

Consider a general weighted sum of a positive definite and symmetric kernel $\sum w_i k_i(\mathbf{x}, \mathbf{x}_i)$. Since we can write k as a dot product in the feature space of it, we can rewrite this expression as $\sum w_i \varphi(\mathbf{x})^T \varphi(\mathbf{x}_i) = \varphi(\mathbf{x})^T \sum w_i \varphi(\mathbf{x}_i)$.

Thus kernel approximation becomes nothing but a dot product of $\varphi(\mathbf{x})$ with a weighted average of the vectors $\varphi(\mathbf{x}_i)$. Similar to our analysis on the heat kernel, we can decompose any arbitrary vector $\varphi(\mathbf{x}_j)$ into a component $\mathbf{d}_j$ in the span of other vectors and a component $\mathbf{o}_j$ orthogonal to the span. Furthermore, $\mathbf{d}_j$ can be written as a linear combination of others as $\mathbf{d}_j = \sum_{i \neq j} a_i^j \varphi(\mathbf{x}_i)$. With these definitions, the expression for the weighted average becomes $\sum_{i \neq j} w_i \varphi(\mathbf{x}_i) + w_j \mathbf{d}_j + w_j \mathbf{o}_j = \sum_{i \neq j} (w_i + w_j a_i^j) \varphi(\mathbf{x}_i) + w_j \mathbf{o}_j$. As proved in Appendix B, $\tilde{s}(\mathbf{x}_j) = \|\mathbf{o}_j\|^2 / \|\varphi(\mathbf{x}_j)\|^2$. If $\tilde{s}(\mathbf{x}_j)$ is small, then we can ignore the term $w_j \mathbf{o}_j$ and the weighted average can be computed by modifying the weights and storing only the vectors $\varphi(\mathbf{x}_i), i \neq j$. Thus small $\tilde{s}(\mathbf{x})$ implies the kernel $k(\mathbf{x}, \mathbf{x}_j)$ can be ignored in the reconstruction.

Another important observation is due to the nature of our approximation of the heat kernel with the normalized adjacency matrix of the graph. The decay of the spectrum of this matrix is directly related to how strongly the underlying graph is connected via the edge weights [Coifman and Lafon 2006]. Since our measure tries

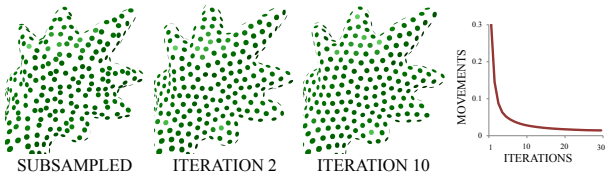

Figure 7: *Points are moved so as to maximize and equalize their contribution to the surface iteratively. Sum of movements of the points decrease quickly.*

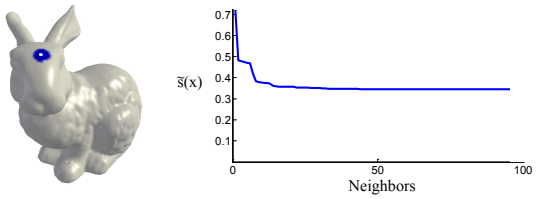

Figure 8: *Using a Gaussian kernel, the measure $\tilde{s}(\mathbf{x})$ for the white point converges to a value as more and more neighboring points are used for the computation. The points in the blue region are sufficient for an almost exact computation, illustrating the local nature of the measure.*

to quantify the change a point makes to the spectrum, it implicitly encodes connectivity of the graph. For kernel reconstruction algorithms to produce accurate results, this connectivity is essential.

These observations lead us to use the same kernel for both the sampling measure $\tilde{s}(\mathbf{x})$ and the reconstruction algorithms. This not only allows to easily reconstruct manifolds and functions from sampled data, but also provides the user with intuitive parameters for the sampling algorithms.

6 Algorithms for Sampling

Having derived our measure and demonstrated how it can be efficiently computed, we move on to algorithms utilizing this measure for sampling. The first algorithm is a simple subsampling algorithm developed for out-of-core and efficient sampling of huge datasets. The second one is a gradient ascent procedure that iteratively moves the points to maximize and equalize their contribution to the manifold. The dense point set is first input to the subsampling algorithm. The output point set is then input to the resampling algorithm. The subsampling algorithm is designed to avoid expensive computations that involves all points in the input point set, making whole sampling very efficient.

6.1 Randomized Linear Scan

Point sets acquired from the real world can be very large and thus out-of-core and fast algorithms are needed to sample these datasets effectively for a given detail level. For this reason, we designed a simple linear scan algorithm. The algorithm starts with an empty output point set. It randomly selects a point $\mathbf{x}_k$ from the input set and computes $\tilde{s}(\mathbf{x}_k)$ using the already added points to the output set. If $\tilde{s}(\mathbf{x}_k) > \epsilon$, then $\mathbf{x}_k$ is added to the output point set. The algorithm stops when all input points have been considered. A pseudo code of the algorithm is given in Algorithm 1.

Algorithm 1 Simplification by Randomized Linear Scan

Input: Initial Point set X
Output: Subsampled point set O

$O = \emptyset$
while $(X \neq \emptyset)$
  remove a random point $\mathbf{x}_k$ from X
  find set of local neighbors N_k of $\mathbf{x}_k$ among the points in O
  if $(N_k \neq \emptyset)$
    compute $\tilde{s}(\mathbf{x}_k)$ (see equation (9)) using points in N_k
  if $(\tilde{s}(\mathbf{x}_k) > \epsilon$ or $N_k = \emptyset)$
    add $\mathbf{x}_k$ to O

Iterative Inversion

The main computational burden of the linear scan algorithm comes from finding local neighbors and inverting the local kernel matrix. To keep this cost at a minimum, we propose to use an iterative algorithm. To compute $\tilde{s}(\mathbf{x}_k)$, first, the points in the neighborhood of $\mathbf{x}_k$ are sorted according to their distance to $\mathbf{x}_k$ in ascending order. This is because points closer to $\mathbf{x}_k$ will contribute to $\tilde{s}(\mathbf{x}_k)$ more. Starting from the closest point, at each iteration, a new point is considered, the new inverse is computed and $\tilde{s}(\mathbf{x}_k)$ is updated. Since every added point decreases $\tilde{s}(\mathbf{x}_k)$ (this is because every added point contributes to the span of the vectors φ, see also Section 4.3), once $\tilde{s}(\mathbf{x}_k) \leq \epsilon$, no further iterations are needed because it is for certain that $\mathbf{x}_k$ will not be added to the output point set.

To compute the inverse iteratively, one can use the block matrix inversion formula that has been proven to be very effective for similar cases [Moghaddam et al. 2008], which reads as follows for our case:

$$\mathbf{K}_{n+1}^{-1} = \begin{bmatrix} \mathbf{K}_n^{-1} + g_n \mathbf{a}_n \mathbf{a}_n^T & -g_n \mathbf{a}_n \\ -g_n \mathbf{a}_n^T & g_n \end{bmatrix} \tag{10}$$

where $\mathbf{K}_n$ is an n by n matrix with elements $(\mathbf{K}_n)_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ for $i, j \leq n$, $\mathbf{a}_n = \mathbf{K}_n^{-1}\mathbf{k}_n(\mathbf{x}_{n+1})$, $g_n = (k(\mathbf{x}_{n+1}, \mathbf{x}_{n+1}) - \mathbf{k}_n(\mathbf{x}_{n+1})^T \mathbf{K}_n^{-1} \mathbf{k}_n(\mathbf{x}_{n+1}))^{-1}$, and $(\mathbf{k}_n(\mathbf{x}_{n+1}))_i = k(\mathbf{x}_{n+1}, \mathbf{x}_i)$ for $i \leq n$.

Note that in this update rule, instabilities arise when g_n^{-1} is close to zero. We can safely avoid these cases by omitting the added point x_{n+1} for which g_n^{-1} is close to zero since this means the mapping of x_{n+1}, $\varphi(x_{n+1})$ (as defined in Section 4.3) is almost in the span of already considered $\varphi(x_i)$'s for $i < n+1$ and thus do not contribute to the projection of $\varphi(x)$ onto this span in computing our measure. This also allows to compute the measure even when the kernel matrix is close to singular.

6.2 Iterative Gradient Ascent

Recall that $\tilde{s}(\mathbf{x})$ measures the contribution of a point to the manifold definition. Hence, by maximizing and equalizing $\tilde{s}(\mathbf{x})$ for all points, we can make sure that each point is contributing equally to the surface. In general, this is a difficult non-linear optimization problem involving modifying positions of points to reach a global optimum. Instead of a global minimization, we use local operations and move points in a simple gradient ascent iteration on $\tilde{s}(\mathbf{x})$. Specifically, at each step, a point $\mathbf{x}$ is selected, moved to the position

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \frac{1}{2} \nabla \tilde{s}(\mathbf{x}^k) \tag{11}$$

and projected onto the surface. The algorithm then continues with the next point randomly chosen among the points that have not been moved. Once all points are exhausted, the algorithm continues with another iteration until a criterion is met.

In practice, this algorithm converges very fast and produces high quality samplings with blue noise properties. An example resampling process is shown in Figure 7. We illustrate and further explain properties of the samplings in Section 7. For reference, one can easily compute the gradient for a Gaussian kernel as $\nabla\tilde{s}(\mathbf{x}) = (-2/\sigma^2)\sum(2\mathbf{x} - \mathbf{x}_i - \mathbf{x}_j)\,k(\mathbf{x},\mathbf{x}_i)k(\mathbf{x},\mathbf{x}_j)\mathbf{K}^{-1}_{ij}$.

6.3 Implementation

Parameters For our simplification algorithm (Algorithm 1), threshold ϵ is the only free parameter apart from the kernel parameters. Since $\tilde{s}(\mathbf{x}) \in [0,1]$, we set $\epsilon = 0.5$ for all results in this paper. For resampling, due to the high convergence rate of the algorithm, we use 10 iterations. Accoring to the decay of the Gaussian $k(\mathbf{x},\mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2}$, the neighborhood size is set to $r = 2.5$.

Data Structures During simplification by linear scan, points are added one by one to the output point set, and the measure computation for a new point is done using only its neighbors among the points in the output set. Thus we need a dynamic data structure that allows local neighbor retrieval. We used a dynamic kd-tree for these reasons. For the iterative gradient ascent algorithm, the data structure should also allow to alter positions of the points already added. Although a grid or octree can be used efficiently for this case, we chose to simply use a kd-tree for computing indices of neighboring points and assume the same neighborhoods in all iterations. This assumption holds in practice since points are already near optimum positions after the subsampling algorithm and only a few iterations are necessary for the resampling algorithm to converge.

6.4 Multiresolution and Progressive Sampling

The linear scan algorithm can trivially be extended to sample progressively similar to EZW encoding [Shapiro 1993], such that a series of point sets P_i with the property that $P_i \subset P_{i+1}, 1 \leq i \leq n$ is obtained. In a coarse-to-fine approach, starting with a large smoothing parameter, a first linear scan is performed and the resulting set is defined as P_1. To get a more detailed version, the smoothing parameter is decreased and another linear scan is performed on the remaining points in the input point set, adding more points to P_1 to get P_2.

One can also start with a small smoothing parameter so as to obtain a point set P_n for a detailed surface, then reduce the parameter and run a linear scan on P_n to subsample it and get P_{n-1}, and continue in this fashion until P_1, the coarsest level is reached. For each P_i, the surface can be reconstructed using the same smoothing parameters as used for subsampling. An example of such a multiresolution reconstruction is shown in Figure 9.

7 Results

To assess the effectiveness of our algorithms, we test the quality of the samplings, accuracy of the reconstructions resulting from the sampled points and performance of the algorithms quantitatively in extensive experiments. For all experiments, models are scaled such that their bounding box has a maximum length of 100.

7.1 Quality of the Samplings

To test the quality of the distributions generated by our sampling algorithms, we first show that they possess high quality blue noise characteristics on a toroidal square. We then show that the same characteristics exist when sampling general surfaces. Finally, we illustrate that the algorithms generate state-of-the-art results when applied to the remeshing problem with well-shaped triangles. Being

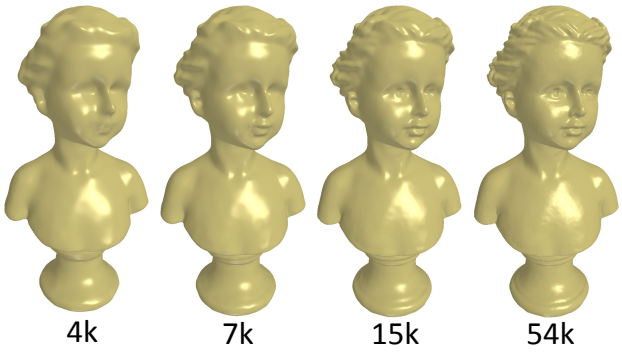

Figure 9: *A multiresolution hierarchy of PSS is obtained by progressive sampling and reconstruction, with indicated number of points.*

Figure 10: *Feature adaptive sampling ($\sigma_p = 0.25$, from left to right: $\sigma_n = \infty$, 0.5).*

out-of-core and efficient, to our knowledge, our algorithm is the first to generate such high quality remeshing results directly from points with little time and space complexity.

Sampling in $\mathbb{R}^2$ One of the special cases of isotropic manifold sampling is sampling the plane or a bounded region in $\mathbb{R}^2$. For this case, quantitative measures to assess the quality of distributions exist. In Figure 11, we show an example distribution on a toroidal square, mean periodogram [Ulichney 1987], power and anisotropy

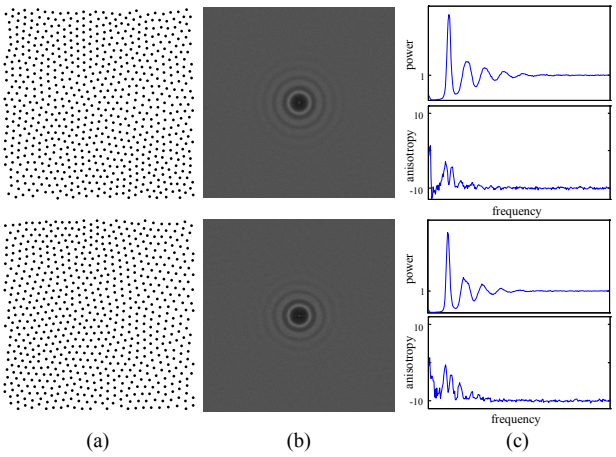

Figure 11: *(a) A distribution of points, (b) mean periodogram, and (c) power and anisotropy graphs for our algorithm (top) and Lloyd's algorithm (bottom).*

plots computed using 10 different random initial distributions for our and Lloyd's (100 iterations) method. The distributions obtained by our algorithms have characteristics similar to Lloyd's method, with an average normalized Poisson disk radius of $\rho = 0.793$. To obtain this distribution, we start with a random sampling of n points and resample by gradient ascent for 10 iterations. The width of the Gaussian kernel is set to $\sigma = 2.5r$, where $r = \rho/\sqrt{(2\sqrt{(3)n})}$ with $\rho = 0.75$ the optimal Poisson disk radius [Lagae and Dutré 2008] to ensure there are enough points in the support of the kernel. We can get the same distributions if we start from a dense sampling of the domain, set a kernel width, subsample and then resample using our algorithms.

Sampling Surfaces We tested our sampling algorithms on different models with different parameters. The parameter σ_p can be tuned to get different smoothness and number of points, and σ_n controls the adaptivity of the samplings. Example samplings are shown in Figure 1, Figure 6 and Figure 10. Setting $\sigma_n = \infty$, one can get uniform sampling of the surface with well-distributed points. Lower values of σ_n causes the algorithms place more samples in the curved regions and features, resulting in preservation of details. As illustrated in Figure 12 and Figure 14, our sampling algorithms are also resilient to noise. Our algorithms depend on both the point set and the kernel to determine the manifold they are working on. Hence, whether a point with noise is considered important or not depends on the smoothness level of the kernel. In Figure 12 (top row), the input to our sampling algorithms contains a point with high normal noise. This point is consistently kept for smaller values of σ_p (middle two figures) and the corresponding bump is present on the circle. For bigger σ_p (rightmost), it is no longer selected as important by the subsampling algorithm and the bump is eliminated.

Remeshing Isotropic distributions are used to remesh surfaces with well-shaped triangles [Yan et al. 2009; Valette et al. 2008] using approximations of the geodesic Centroidal Voronoi Diagram [Du et al. 1999]. To quantitatively measure quality of our samplings of surfaces, we compare meshes obtained by using sampled points output from our algorithms as vertices to high quality remeshing algorithms. We use Tight Cocone [Dey and Goswami 2003] to triangulate the samples and compare triangle qualities to those generated by a fast clustering based approach (VR) [Valette et al. 2008] and a high quality triangle producing but slower method (YR) [Yan et al. 2009].

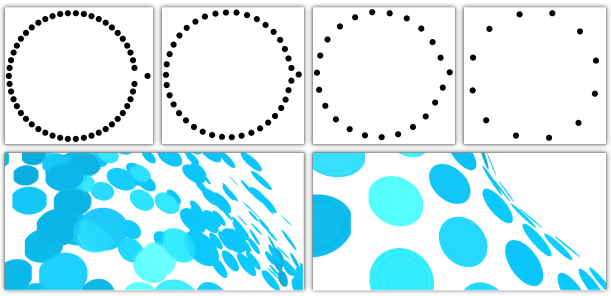

Figure 12: *Top: Input to our algorithms contains one point with high normal noise (leftmost). Resamplings are shown for increasing σ_p. Bottom: A noisy surface part and its resampling.*

We use the same measures as used in the papers [Yan et al. 2009; Valette et al. 2008]. θ_{min} is the smallest angle in the mesh, $\theta_{min,ave}$ is the average of minimum angles of all triangles, and $\theta < 30^o$ is the percentage of triangles with minimum angle below 30^o. Q denotes quality of a triangle and measured as $6S_t/\left(\sqrt{3}p_t h_t\right)$, where S_t is the area of the triangle t, p_t is its half-perimeter, and h_t is the length of the longest edge. Q_{min} and Q_{ave} are the minimum and average of Q's of all triangles in the mesh. All measures for the models in the figures are provided in Table 1.

As can be observed from Table 1, our algorithms run in comparable times to VR but still provide triangles with qualities similar to YR. Owing to the inherent smoothing of our algorithms, it also works for very noisy cases without pre-smoothing as illustrated in Figure 14 (and the corresponding entry in Table 1). The triangle quality is higher and geometry is well-captured. The blue noise characteristics of the distribution of the points generated by our algorithms result in more non-regular triangles as can be seen in Figure 13. Furthermore, since our algorithms are out-of-core and very efficient, they can be applied to very large datasets such as the Lucy model with 14 million points (Table 1).

7.2 Surface Reconstruction

We test the accuracy of the reconstructions by direct comparisons with two point-based iterative simplification algorithms which remove points according to their distance to the surface (AS) [Alexa et al. 2001] or a kernel based measure (KS) [Kitago and Gopi 2006]. Reconstructions using the initial dense point set and the simplified point sets are compared in terms of the root mean square (RMS) error and Hausdorff distance. Note that we run only our simplification algorithm and not the resampling for fair comparisons.

We use a variety of models of different complexity, genus, and source and a range of σ_p values to illustrate the quality of the reconstructions under different conditions. The parameter σ_n is set to 0.75 to provide adaptive sampling [Öztireli et al. 2009]. After densely triangulating the implicit function we use for reconstructions, Metro tool [Cignoni et al. 2001] is run for computing the errors. We plot the results of our tests in Table 2 and show examples of reconstructions obtained in Figure 15. Our simplification algorithm results in more accurate reconstructions, with considerable improvements especially for complex models. In Figure 15 bottom row, the reconstructions using the simplified point sets of AS and KS result in extra surface parts and distortions on the surface while our algorithm almost exactly reproduces the original reconstruction from all points.

7.3 Performance

Linear scanning of the input points avoids costly operations such as finding neighbors and taking local kernel matrix inverses among all input points. This makes our algorithms run in comparable times to even mesh-based subsampling methods. Point based simplification algorithms AS and KS have much larger time complexity and become infeasible to use for large models and large σ_p.

Performance of our algorithms is illustrated in Figure 16. In Figure 16 (a), total time (including kd-tree queries and disk reads) needed to subsample models of various sizes is plotted. Complexity grows linearly with input size due to the linear scan used in the subsampling algorithm. Figure 16 (b) illustrates the same performance analysis for the resampling algorithm. Note that the input to this algorithm is output by the subsampling algorithm and thus number of neighbors to be considered for the kernel matrix inversions stays approximately constant. In Figure 16 (c), number of output points, hence σ_p, is changed for the subsampling algorithm. Since we use iterative inversion (as explained in Section 6.1) for computing our measure, iterations are cut earlier if σ_p is large and the complexity stays constant.

8 Conclusions

We presented new algorithms for simplification and resampling of manifolds. The algorithms depend on a measure that restricts changes to the Laplace-Beltrami spectrum. By utilizing kernel methods and matrix perturbation theory, we were able to derive a local measure for efficient sampling. We then utilized this measure in an efficient, out-of-core and accurate simplification algorithm. Using the same measure, we showed that resampling can be achieved by simple gradient ascent on the simplified point set. The generated samplings have high quality isotropic characteristics and result in accurate reconstructions.

Limitations Although our algorithms are efficient and accurate, they are greedy and thus not theoretically guaranteed to give the optimal sampling. We believe however, that our analysis of the Laplace-Beltrami spectrum can be utilized for more sophisticated algorithms as well. Our remsampling algorithm is very effective with high convergence rates when used after applying our simplification algorithm to the data. However, if it is directly used on a point set with a very non-uniform distribution, limitations on convergence and stability may arise.

Future Directions The current sampling scheme can be adapted to settings where a function such as texture on a surface needs to be faithfully reconstructed. This can be easily achieved by adding new dimensions to the space where the kernel operates, corresponding to values derived from the function. By utilizing local feature size, smoothly degrading triangles and isotropic adaptive remeshing can be achieved. Application of our algorithms to high dimensional datasets that are assumed to have a low dimensional structure could speed up existing methods. Local resampling can also be supported by selecting a local patch and keeping the patch boundaries fixed. For efficiency, upper bounds on the eigenvalues can be incorporated to avoid as many matrix inversions as possible. Also, inversion and linear system solving can be programmed on the GPU.

Sampling is ultimately related to signal processing and multiresolution analysis. Computational harmonic analysis gives us a unified framework to perform multiresolution analysis on general manifolds. We presented an effective application of it to sampling. We believe that these and similar ideas from irregular sampling, sparse coding, and machine learning will be useful to understand the sampling problem better and develop practical algorithms with theoretical guarantees.

Acknowledgments

We would like to thank the reviewers for their insightful comments and suggestions, and Gaël Guennebaud for his "Expe" application. The models used in the paper are courtesy of AIM@SHAPE Shape Repository, Stanford University Computer Graphics Laboratory, Cyberware, SensAble, MIT CSAIL, INRIA, IMATI and Clemson University.

References

ACHLIOPTAS, D., MCSHERRY, F., AND SCHÖLKOPF, B. 2001. Sampling techniques for kernel methods. In *NIPS*, 335–342.

ALEXA, M., BEHR, J., COHEN-OR, D., FLEISHMAN, S., LEVIN, D., AND SILVA, C. T. 2001. Point set surfaces. In *VIS '01*, IEEE Computer Society, Washington, DC, USA, 21–28.

ALEXA, M., BEHR, J., COHEN-OR, D., FLEISHMAN, S., LEVIN, D., AND SILVA, C. T. 2003. Computing and rendering point set surfaces. *IEEE Transactions on Computer Graphics and Visualization 9*, 1, 3–15.

ALLIEZ, P., MEYER, M., AND DESBRUN, M. 2002. Interactive geometry remeshing. *ACM Trans. Graph. 21*, 3, 347–354.

ALLIEZ, P., VERDIÈRE, E. C. D., DEVILLERS, O., AND ISENBURG, M. 2003. Isotropic surface remeshing. In *SMI '03*, IEEE Computer Society, Washington, DC, USA, 49.

BAUER, U., POLTHIER, K., AND WARDETZKY, M., 2009. Uniform convergence of discrete curvatures from nets of curvature lines, Dec. 14.

BELKIN, M., AND NIYOGI, P. 2006. Convergence of laplacian eigenmaps. In *NIPS*, 129–136.

BOBENKO, A., AND SPRINGBORN, B. 2007. A discrete laplace-beltrami operator for simplicial surfaces. *Discr. Comp. Geom 38*, 4, 740–756.

BOISSONNAT, J. D., AND OUDOT, S. 2003. Provably good surface sampling and approximation. In *SGP '03*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 9–18.

BRAUN, M. L. 2006. Accurate error bounds for the eigenvalues of the kernel matrix. *J. Mach. Learn. Res. 7*, 2303–2328.

CARR, J. C., BEATSON, R. K., CHERRIE, J. B., MITCHELL, T. J., FRIGHT, W. R., MCCALLUM, B. C., AND EVANS, T. R. 2001. Reconstruction and representation of 3d objects with radial basis functions. In *SIGGRAPH 2001: Proc. of the 28th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 67–76.

CHENG, S.-Y. 1976. Eigenfunctions and nodal sets. *Commentarii Mathematici Helvetici 51*, 1, 43–55.

CIGNONI, P., ROCCHINI, C., AND SCOPIGNO, R. 2001. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum 17*, 2, 167–174.

COHEN-STEINER, D., AND MORVAN, J.-M. 2006. Second fundamental measure of geometric sets and local approx- imation of curvatures. *J. Differential Geom. 73*, 3, 363–394.

COIFMAN, R. R., AND LAFON, S. 2006. Diffusion maps. *Applied and Computational Harmonic Analysis 21*, 1, 5–30.

COOK, R. L. 1986. Stochastic sampling in computer graphics. *ACM Trans. Graph. 5*, 1, 51–72.

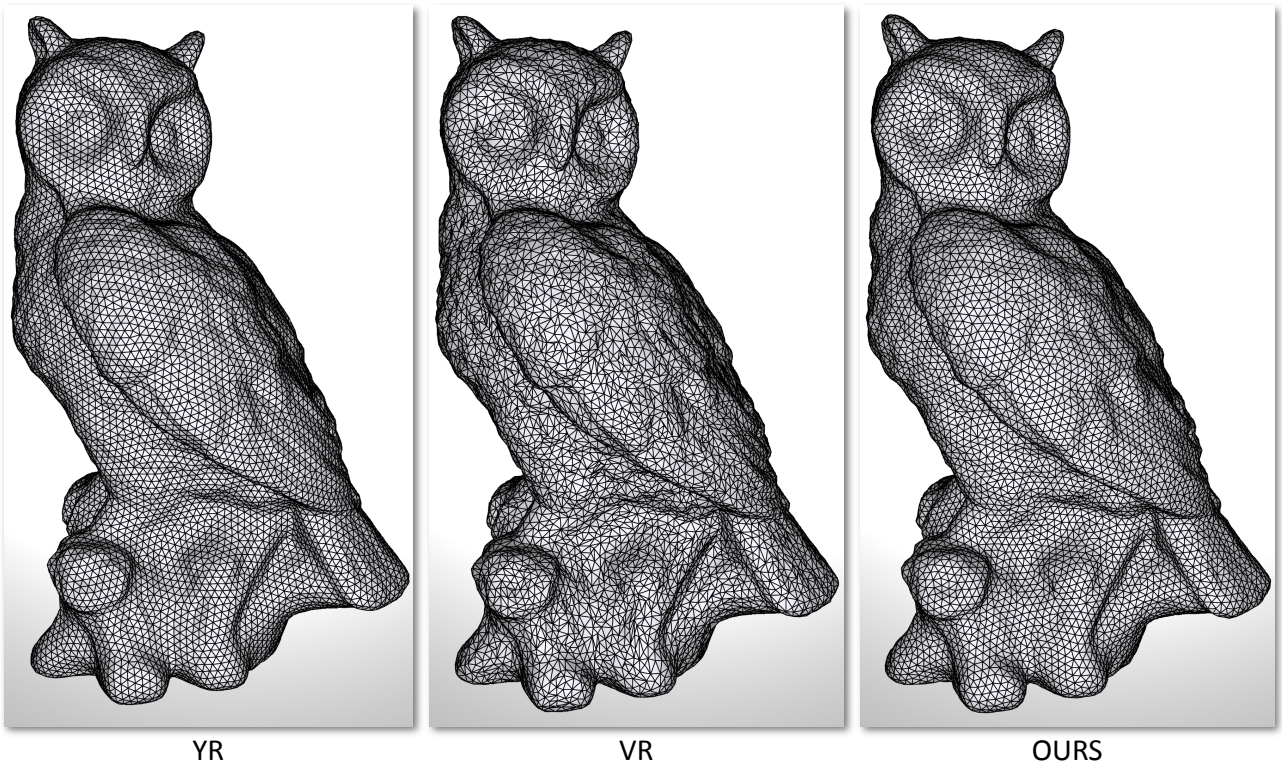

Figure 13: *Remeshing results for the Owl model.*

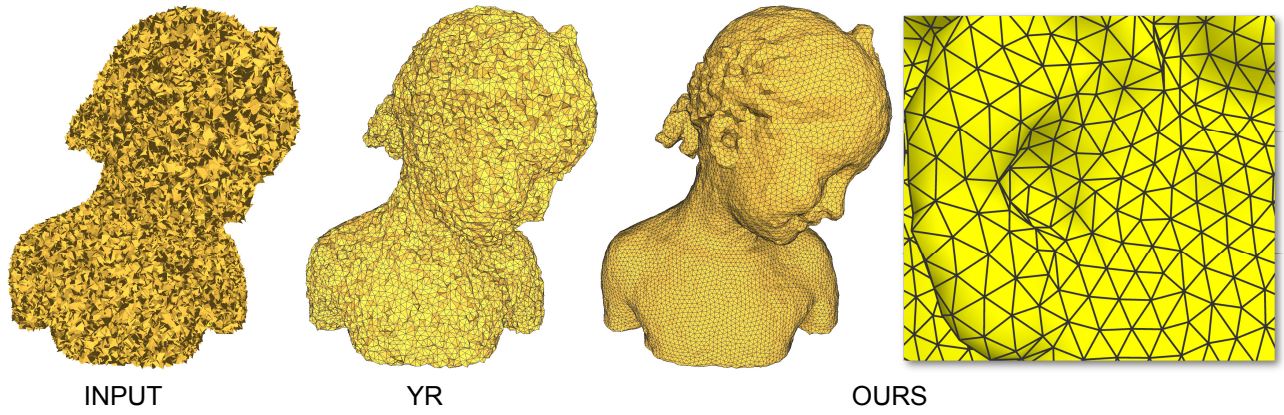

Figure 14: *Remeshing results for the Bimba model with YR and our algorithm. Due to the inherent smoothing of our approach, the geometric shape is well-captured while achieving high triangle quality without pre-processing.*

DEY, T. K., AND GOSWAMI, S. 2003. Tight cocone: a water-tight surface reconstructor. In *SM '03*, ACM, New York, NY, USA, 127–134.

DEY, T. K., RAJAN, P., AND WANG, Y. 2010. Convergence, stability, and discrete approximation of laplace spectra. In *SODA '10*, ACM. to appear.

DRINEAS, P., AND MAHONEY, M. W. 2005. Approximating a gram matrix for improved kernel-based learning. In *Learning Theory*, vol. 3559. Springer Berlin / Heidelberg, 323–337.

DU, Q., FABER, V., AND GUNZBURGER, M. 1999. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review 41*, 4, 637–676.

FU, Y., AND ZHOU, B. 2008. Direct sampling on surfaces for high quality remeshing. In *SPM '08*, ACM, New York, NY, USA, 115–124.

GARLAND, M., AND HECKBERT, P. S. 1997. Surface simplification using quadric error metrics. In *SIGGRAPH 97: Proc. of the 24th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 209–216.

GRIGOR'YAN, A. 1998. Estimates of heat kernels on riemannian manifolds. In *Spectral Theory and Geometry. ICMS Instructional Conference*, Cambridge Univ. Press, 140–225.

GUENNEBAUD, G., AND GROSS, M. 2007. Algebraic point set

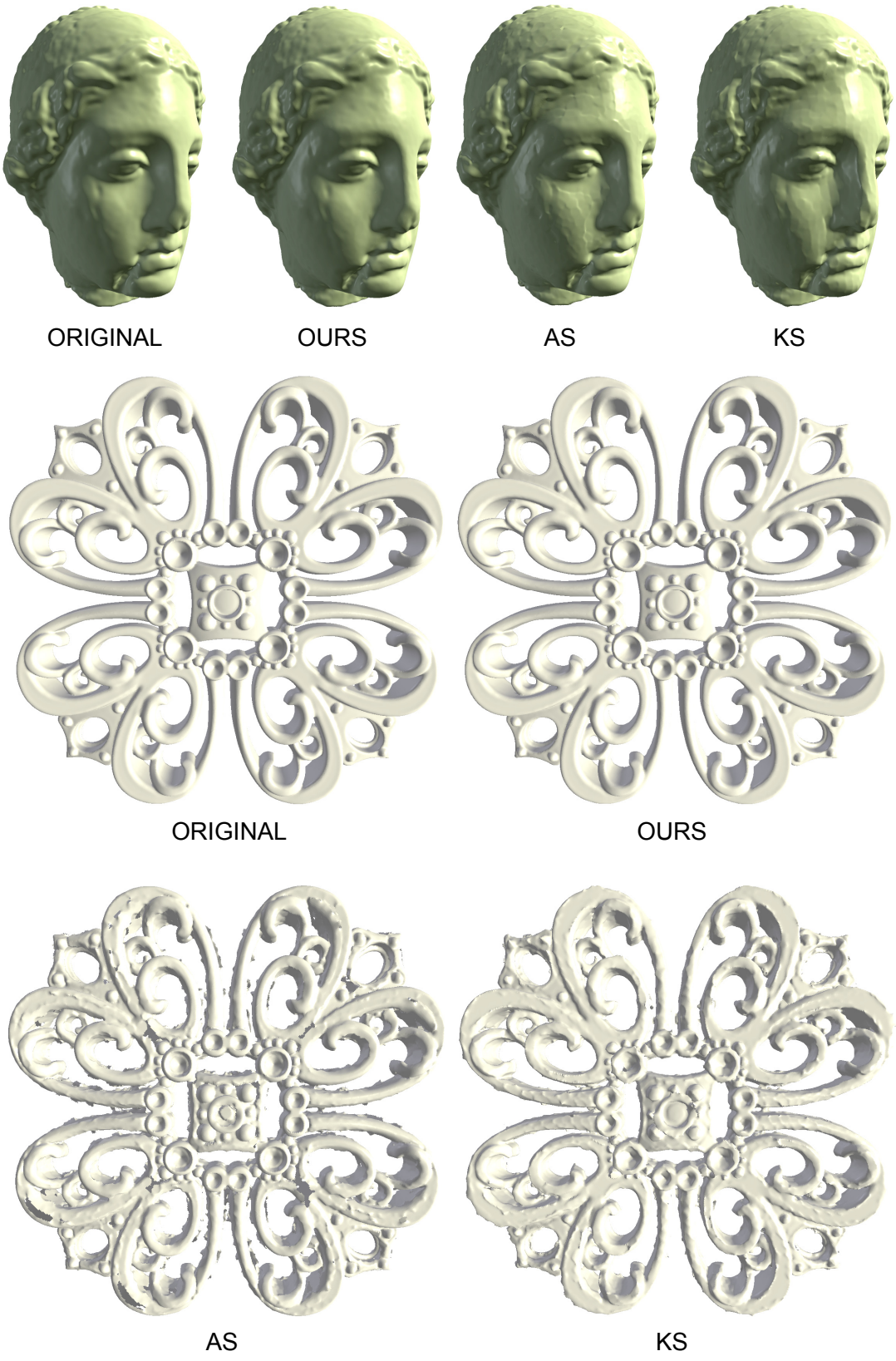

Figure 15: *Reconstructions of the Igea (top row, $\sigma_p = 0.75$, $\sigma_n = 0.75$) and Filigree ($\sigma_p = 0.5$, $\sigma_n = 0.75$) models using the full data, and subsampled data output from different algorithms.*

Model	Num Pts	Method	θ_{min}	$\theta_{min,\ ave}$	Q_{min}	Q_{ave}	$\theta < 30^o$	Time
Owl	13161	**Ours**	**19.34**	**52.62**	**0.3007**	**0.8998**	**0.0012**	**3.37 + 6.51**
		VR	0.61	39.19	0.0112	0.7194	6.3025	4.453
		YR	37.35	54.71	0.6548	0.9371	0	319
Horse	8600	**Ours**	**32.52**	**53.01**	**0.5844**	**0.9057**	**0**	**2.25 + 4.18**
		VR	14.76	46.24	0.3148	0.8250	0.3372	3.579
		YR	37.92	54.84	0.6562	0.9389	0	290
Rabbit	9635	**Ours**	**35.19**	**53.02**	**0.5573**	**0.9054**	**0**	**2.50 + 4.65**
		VR	10.69	44.67	0.2138	0.8036	1.1747	4.859
		YR	39.04	55.09	0.6649	0.9414	0	255
Bimba	13318	**Ours**	**17.10**	**52.68**	**0.2664**	**0.9016**	**0.0003**	**1.109 + 2.61**
		VR	0.91	37.73	0.0144	0.6993	0.0948	4.031
		YR	20.13	46.17	0.3722	0.8266	0.0030	1424
Lucy	3259	**Ours**	**13.58**	**51.81**	**0.2727**	**0.8915**	**0.2222**	**367.56 + 1.46**
	7395	**Ours**	**11.14**	**51.74**	**0.2538**	**0.8906**	**0.3547**	**367.29 + 3.79**
C. Lion	38342	**Ours**	**14.56**	**52.83**	**0.2216**	**0.9033**	**0.0082**	**18.83 + 22.58**

Table 1: *Quantitative comparisons of the remeshing results. Timings for our method, as well as for Tight Cocone is given in the rows for our algorithm. "Num Pts" refers to the number of vertices in the output mesh.*

		$\sigma_p = 0.5$			$\sigma_p = 0.75$			$\sigma_p = 1$		
Model	Error Metric	**Ours**	AS	KS	**Ours**	AS	KS	**Ours**	AS	KS
Igea	RMS	**0.0071**	0.0064	0.0091	**0.0114**	0.0310	0.0818	**0.0407**	0.0824	0.1251
	HAUS	**0.1075**	0.1415	0.1120	**0.1500**	0.4384	0.4640	**0.1907**	1.3302	2.0278
Hand	RMS	**0.0186**	0.1628	0.1464	**0.0286**	0.5405	0.5873	**0.0439**	1.0162	1.0382
	HAUS	**0.2116**	1.2040	1.1011	**0.2472**	2.0007	2.1906	**0.5616**	2.9348	2.7861
Statue	RMS	**0.0488**	0.2357	0.2796	**0.0949**	0.5980	0.6975	**0.1501**	1.0184	0.7677
	HAUS	**1.2853**	1.3792	1.4335	**1.9832**	2.0409	2.4409	**2.5519**	3.7673	2.5541
Dragon	RMS	**0.0526**	0.1011	0.3299	**0.0973**	0.2514	0.6862	**0.1495**	0.5747	1.4733
	HAUS	**1.4130**	1.3971	1.7840	**2.0595**	2.0062	2.6382	**2.6049**	2.8164	4.3328
Filigree	RMS	**0.0209**	0.3206	0.1623	**0.0457**	0.8481	0.4929	**0.0832**	1.3032	1.0696
	HAUS	**0.2591**	1.3294	1.1363	**0.4363**	2.2062	1.7476	**0.8408**	2.6989	2.5756

Table 2: *Quantitative comparisons of the reconstructions.*

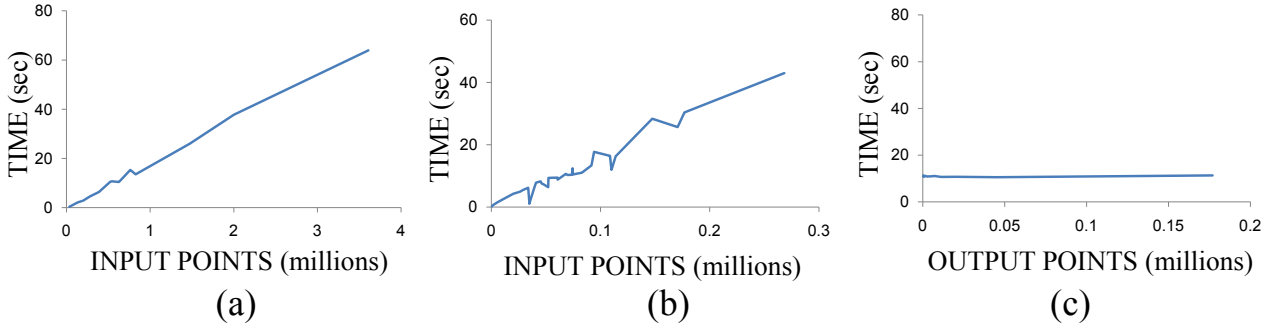

Figure 16: *Effect of the input model size on the time complexity of the (a) subsampling algorithm and (b) resampling algorithm. (c) Timings for the subsampling algorithm for a fixed model with changing output sample sizes.*

surfaces. *ACM Trans. Graph. (SIGGRAPH 2007) 26*, 3, 23.1–23.9.

HAM, J., LEE, D. D., MIKA, S., AND SCHÖLKOPF, B. 2004. A kernel view of the dimensionality reduction of manifolds. In

ICML '04, ACM, New York, NY, USA, 47.

IPSEN, I., AND NADLER, B. 2009. Refined perturbation bounds for eigenvalues of hermitian and non-hermitian matrices. *SIAM Journal of Matrix Analysis 31*, 1, 40–53.

KAROUI, N. E., AND D'ASPREMONT, A. 2009. Approximating eigenvectors by subsampling. Tech. Rep. arXiv:0908.0137, Aug.

KESAVAN, S. 1998. Listening to the shape of a drum. *Resonance 3*, 49–58. 10.1007/BF02841422.

KITAGO, M., AND GOPI, M. 2006. Efficient and prioritized point subsampling for csrbf compression. In *Symp. on Point-based Graphics, Eurographics*.

LAFON, S., AND LEE, A. 2006. Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 28*, 9 (Sept.), 1393–1403.

LAGAE, A., AND DUTRÉ, P. 2008. A comparison of methods for generating Poisson disk distributions. *Computer Graphics Forum 27*, 1 (March), 114–129.

LAI, Y.-K., ZHOU, Q.-Y., HU, S.-M., WALLNER, J., AND POTTMANN, H. 2007. Robust feature classification and editing. *IEEE Trans. Vis. Comp. Graphics 13*, 1, 34–45.

LEVIN, D. 2003. Mesh-independent surface interpolation. *Geometric Modeling for Scientific Visualization*, 37–49.

LÉVY, B. 2006. Laplace-beltrami eigenfunctions towards an algorithm that "understands" geometry. In *Shape Modeling International*, IEEE Computer Society, 13.

LIU, R., JAIN, V., AND ZHANG, H. 2006. Subsampling for efficient spectral mesh processing. In *Computer Graphics International*, 172–184.

MOGHADDAM, B., GRUBER, A., WEISS, Y., AND AVIDAN, S. 2008. Sparse regression as a sparse eigenvalue problem. In *Information Theory and Applications Workshop, 2008*, 121–127.

OHTAKE, Y., BELYAEV, A., AND SEIDEL, H.-P. 2004. 3d scattered data approximation with adaptive compactly supported radial basis functions. In *Shape Modeling Applications*, 31–39.

ÖZTIRELI, C., GUENNEBAUD, G., AND GROSS, M. 2009. Feature preserving point set surfaces based on non-linear kernel regression. In *Eurographics 2009*, 493–501.

PAULY, M., AND GROSS, M. 2001. Spectral processing of point-sampled geometry. In *SIGGRAPH 2001: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 379–386.

PAULY, M., GROSS, M., AND KOBBELT, L. P. 2002. Efficient simplification of point-sampled surfaces. In *VIS '02*, IEEE Computer Society, Washington, DC, USA, 163–170.

PEYRÉ, G., AND COHEN, L. D. 2006. Geodesic remeshing using front propagation. *Int. J. Comput. Vision 69*, 1, 145–156.

PINKALL, U., AND POLTHIER, K. 1993. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics 2*, 1, 15–36.

REUTER, M., WOLTER, F.-E., AND PEINECKE, N. 2006. Laplace-beltrami spectra as "shape-dna" of surfaces and solids. *Computer-Aided Design 38*, 4, 342–366.

RUSTAMOV, R. M. 2007. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *SGP07*, Eurographics Association, Barcelona, Spain, A. Belyaev and M. Garland, Eds., 225–233.

SAUCAN, E., APPLEBOIM, E., AND ZEEVI, Y. Y. 2008. Sampling and reconstruction of surfaces and higher dimensional manifolds. *J. Math. Imaging Vis. 30*, 1, 105–123.

SCHÖLKOPF, B., SMOLA, A., AND MÜLLER, K.-R. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput. 10*, 5, 1299–1319.

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

SHAPIRO, J. M. 1993. Embedded image coding using zerotrees of wavelet coefficients. *Signal Processing, IEEE Transactions on 41*, 12, 3445–3462.

SUN, J., OVSJANIKOV, M., AND GUIBAS, L. J. 2009. A concise and provably informative multi-scale signature based on heat diffusion. *Comput. Graph. Forum 28*, 5, 1383–1392.

TAUBIN, G. 1995. A signal processing approach to fair surface design. In *SIGGRAPH 95: Proc. of the 22nd annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 351–358.

TURK, G. 1992. Re-tiling polygonal surfaces. *SIGGRAPH Comput. Graph. 26*, 2, 55–64.

ULICHNEY, R. 1987. *Digital halftoning*. MIT Press, Cambridge, MA, USA.

VALETTE, S., CHASSERY, J. M., AND PROST, R. 2008. Generic remeshing of 3d triangular meshes with metric-dependent discrete voronoi diagrams. *IEEE Transactions on Visualization and Computer Graphics 14*, 369–381.

VALLET, B., AND LÉVY, B. 2008. Spectral geometry processing with manifold harmonics. *Computer Graphics Forum (Eurographics)*.

VON LUXBURG, U. 2004. *Statistical Learning with Similarity and Dissimilarity Functions*. PhD thesis, Technischen Universität Berlin.

WARDETZKY, M., MATHUR, S., KÄLBERER, F., AND GRINSPUN, E. 2007. Discrete laplace operators: no free lunch. In *SGP '07*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 33–37.

WITKIN, A., AND HECKBERT, P. S. 1994. Using particles to sample and control implicit surfaces. In *21st annual conference on Computer graphics and interactive techniques*, ACM Press, 269–278.

YAN, D.-M., LÉVY, B., LIU, Y., SUN, F., AND WANG, W. 2009. Isotropic remeshing with fast and exact computation of restricted voronoi diagram. *Comput. Graph. Forum (Symp. on Geometry Processing 2009) 28*, 5, 1445–1455.

ZHANG, H., VAN KAICK, O., AND DYER, R. 2007. Spectral methods for mesh processing and analysis. In *Eurographics State-of-the-art Report*, 1–22.

A Distribution of Effect of a Point on the Spectrum

Suppose that the eigenvectors $\mathbf{u}_i$ of the $n \times n$ matrix $\mathbf{H}$ sample the eigenfunctions u_i such that $(\mathbf{u}_i)_j = u_i(x_j)$. Then eigenvalues of $\mathbf{H}$ also well approximate those of h [Braun 2006], thus we can approximately set $\lambda_i(\mathbf{H}) = e^{-\lambda_i t}$. If we define the eigenvector matrix as $\mathbf{U}^T_{ij} = (\mathbf{u}_i)_j$, it can be shown [Schölkopf et al. 1998] that there is the following relation between $\mathbf{v}_i$ and $\mathbf{U}$:

$$\mathbf{v}_i = \frac{1}{\sqrt{\lambda_i(\mathbf{H})}} \sum_{d=1}^{n} \mathbf{U}^T_{id} \phi_d$$

Hence, we can write

$$\begin{aligned}
\delta_i = (\phi^T \mathbf{v}_i)^2 &= \frac{1}{\lambda_i(\mathbf{H})} \left(\sum_{d=1}^{n} \mathbf{U}^T_{id} (\phi^T \phi_d) \right)^2 \\
&= \frac{1}{e^{-\lambda_i t}} \left(\sum_{d=1}^{n} \mathbf{U}^T_{id} h(x, x_d) \right)^2 \\
&= \frac{1}{e^{-\lambda_i t}} \left(\sum_{d=1}^{n} \mathbf{U}^T_{id} \sum_{k=0}^{\infty} e^{-\lambda_k t} u_k(x) u_k(x_d) \right)^2 \\
&= \frac{1}{e^{-\lambda_i t}} \left(\sum_{k=0}^{\infty} e^{-\lambda_k t} u_k(x) \sum_{d=1}^{n} \mathbf{U}^T_{id} \mathbf{U}_{dk} \right)^2 \\
&= \frac{1}{e^{-\lambda_i t}} \left(e^{-\lambda_i t} u_i(x) \right)^2 \\
&= e^{-\lambda_i t} u_i^2(x)
\end{aligned}$$

In the fourth step, we used the fact that $\mathbf{U}$ is a unitary matrix. This means that as long as $u_i(x)$'s are not simultenously zero and for short time, each δ_i will get a non-zero value. The zero set of $u_i(x)$ is called the nodal set (for 2-manifolds, the nodal line) with dimension one less than the manifold and can have at most $i + 1$ nodal domains, where its sign does not change, for λ_i in increasing order [Cheng 1976]. This implies that those points where more than one $u_i(x)$ is zero should be at the interesection of many nodal sets. Hence, for most points on the maniold, most δ_i's will be non-zero.

B Projection onto a Span in a Reproducing Kernel Hilbert Space

Suppose we have vectors $\phi_i = \phi(\mathbf{x}_i)$ that form the span, and the vector $\phi = \phi(\mathbf{x})$ to be projected onto the span. We also assume that dot product in this vector space is given by the kernel k such that $k(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^T \phi(\mathbf{y})$. Although it is possible to find a basis of the span and compute the norm of the projection, an easier way is using the fact that projection of ϕ is the closest point of ϕ on the span. Thus it is sufficient to minimize the distance of ϕ to the span to compute the norm of the orthogonal component $\|\mathbf{o}\|$, from which norm of the projection follows as $\|\mathbf{d}\|^2 = \|\phi\|^2 - \|\mathbf{o}\|^2$.

We want to minimize the distance of ϕ to the span, which is given by $\|\phi - \sum a_i \phi_i\|^2$ with respect to a_i's. Denoting the kernel matrix as $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, and defining the vectors $(\mathbf{a})_i = a_i$ and $(\mathbf{k})_i = k(\mathbf{x}, \mathbf{x}_i)$, this minimization can be expressed as

$$\|\phi\|^2 + \|\sum a_i \phi_i\|^2 - 2\phi^T \sum a_i \phi_i = k(\mathbf{x}, \mathbf{x}) + \mathbf{a}^T \mathbf{K} \mathbf{a} - 2\mathbf{k}^T \mathbf{a}$$

which is minimized for $\mathbf{a} = \mathbf{K}^{-1}\mathbf{k}$. Substituting this expression into $\|\phi\|^2 - \|\phi - \sum a_i \phi_i\|^2$ we get $\|\mathbf{d}\|^2 = \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k}$.