Integral Estimation from Point Cloud in d-Dimensional Space: A Geometric View

Chuanjiang Luo* Jian Sun† Yusu Wang*

Abstract

Integration over a domain, such as a Euclidean space or a Riemannian manifold, is a fundamental problem across scientific fields. Many times, the underlying domain is only accessible through a discrete approximation, such as a set of points sampled from it, and it is crucial to be able to estimate integral in such discrete settings. In this paper, we study the problem of estimating the integral of a function defined over a k-submanifold embedded in $\mathbb{R}^d$, from its function values at a set of sample points. Previously, such estimation is usually obtained in a statistical setting, where input data is typically assumed to be drawn from certain probabilistic distribution. Our paper is the first to consider this important problem of estimating integral from point clouds data (PCD) under the more general non-statistical setting, and provide certain theoretical guarantees.

Our approaches consider the problem from a geometric point of view. Specifically, we estimate the integral by computing a weighted sum, and propose two weighting schemes: the *Voronoi* and the *Principal Eigenvector* schemes. The running time of both methods depends mostly on the intrinsic dimension of the underlying manifold, instead of on the ambient dimensions. We show that the estimation based on the Voronoi scheme converges to the true integral under the so-called (ε, δ)-sampling condition with explicit error bound presented. This is the first result of this sort for estimating integral from general PCD. For the Principal Eigenvector scheme, although no theoretical guarantee is established, we present its connection to the Heat diffusion operator, and illustrate justifications behind its construction. Experimental results show that both new methods consistently produce more accurate integral estimations than common statistical methods under various sampling conditions.

1 Introduction

Integration over a domain, such as a Euclidean space or a Riemannian manifold, is a fundamental problem across a broad range of scientific areas. Its importance is perhaps best reflected by the fact that it is one of the most intensively investigated topics in numerical analysis. Indeed, data analysis routinely boils down to collecting certain properties over a domain of interest. For example, in Bayesian net, integration is one of the main components behind information inference. In graphics and visualization, histograms, typically obtained by integrating certain quantities over a domain, are widely used to characterize input shapes.

In many applications, the underlying domains are only accessible through some discrete approximations, most often through a set of sampled points; and the function to be integrated is only given by values at these points. For example, in computer graphics, a physical object may be digitized by using 3-D scanning equipments. In sensor networks, quantities such as temperatures

*Computer Science and Engineering Department, The Ohio State University, Columbus, OH 43210.
†Computer Science Department, Stanford University, Palo Alto CA 94305.

or humidity are usually collected at each individual sensor. To address the integration problem arisen in these common scenarios, in this paper, we study the problem of estimating the integral of an input function from a point clouds data (PCD). We assume that the input data reside on a low-dimensional manifold (while the dimension of its embedded space may be high), which is believed to largely hold for many practical applications. One standard example of such data is the space of images of an object taken under fixed lighting conditions with a rotating camera. While the dimension of the ambient space is proportional to the number of pixels which can be millions, the intrinsic dimension of the image space is only two (i.e, rotation angles of the camera).

Integral estimation from point clouds also falls into the broad category of information recovery and data analysis over low-dimensional manifold structure from point clouds, which has recently received great attention from many research areas, including machine learning and computational geometry [1, 3, 8, 19]. Reconstructing manifold structure is typically costly for high dimensional data, and many times not necessary. Our goal in this paper is to estimate integral information faithfully from the point clouds data without reconstructing the entire underlying manifold structure.

Prior work. Previously, integral estimation from point clouds is usually conducted in a statistical setting, where the input point samples are assumed to be drawn from certain probability distribution. The most popular method to estimate integral in such case is the Monte Carlo integration [16]. In its simplest form, this method simply estimates the integral as the average of the function values at sample points under the assumption that the samples are uniformly distributed. In the case where the uniform assumption is not valid, one can use density estimation technique to obtain statistically guaranteed results. The theoretical guarantee (i.e, closeness to the ground truth) of the Monte Carlo integration method is usually derived based on the *Law of Large Numbers*. Note that the Monto Carlo scheme only estimates the integral up to a scaling factor, which is the volume (surface volume) of the manifold. Volume estimation (which can be considered as the integral of the constant function) is itself a hard problem in high dimensions [14].

It is often much easier to sample the ambient space $\mathbb{R}^d$ according to some distribution than sampling a submanifold of it, even if the submanifold is given explicitly. Li et al. [20] convert the estimation of the area of a surface in $\mathbb{R}^3$ to the estimation of the number of the intersections of lines in $\mathbb{R}^3$ with this surface, based on the the so-called Cauchy-Crofton formula [23]. Since a line in $\mathbb{R}^3$ has four independent parameters, one can uniformly sample the space of lines in $\mathbb{R}^3$ by a uniform sampling of this 4-dimensional line space. Liu et al. [21] adapt this idea to estimate the surface area directly from sample points. However, it is not clear how to extend this strategy to the high dimensional case since the co-dimension of the submanifold in $\mathbb{R}^d$ may be much bigger than one, whence the probability of a line intersecting the submanifold is zero.

From a geometric point of view, the most natural way to estimate integral from PCD is perhaps to first reconstruct a mesh from the sample points which approximates the (metric of the) underlying submanifold. Indeed, once a mesh approximation is given, it is shown in [6] that one can then use the area of the mesh elements incident to each point as the weight for this point, and compute the integral as a weighted sum. However, although efficient algorithms for converting a point cloud from surface in $\mathbb{R}^3$ into a mesh have been developed [1, 2, 11], the mesh construction problem is rather expensive in high dimensions. The best existing such algorithm [8] takes time exponential in the dimension of the ambient space, which, in most of the applications, is much higher than the intrinsic dimension of the manifold. Hence it is highly desirable to have a scheme that directly operates on the sample points, with running time depending only mildly on the ambient dimension, which our paper aims at developing.

Our contribution. In this paper, we consider the problem of integral estimation from PCD in a non-statistical setting where input data are not necessarily randomly sampled. We approach it from a geometric point view, and develop two new algorithms that operate on the PCD directly, with running time mostly depending on the intrinsic dimension of the underlying manifold. Specifically, given a point cloud P sampled from a k-dimensional manifold M embedded in $\mathbb{R}^d$, we approximate the integral of f over M by a weighted sum $\sum_{p\in P} w(p)f(p)$, and propose two novel weighting schemes to compute $w(p)$s. Both new schemes rely on local information from each input point, and can be implemented in time exponential only in the intrinsic dimension of M, which is in some sense the best one can hope for in the deterministic setting [13]. These are the first results of this sort for this important problem in a general setting.

Our first scheme is called the *Voronoi weighting scheme*. It reconstructs a local patch around each input point, and then computes the weight of this point based on it. Although these local patches are not consistent and can not be stitched together to form a global mesh approximating the underlying manifold, we show that they approximate the volume measure of the underlying manifold, and thus produce a convergent estimation under some mild assumption on the sample points and the integrand. We present explicit error bound of our estimation depending on the sampling conditions of input points.

The second scheme is called the *Principal Eigenvector weighting scheme*. The method falls into the same framework as the recent development in constructing Laplace operators from PCD. Although we are not able to establish the convergent result for this scheme, it preserves the global property that Heat diffusion on manifolds is an integral invariant process in the discrete setting, which is an important property given that the Heat diffusion has been widely used to smooth functions in many application areas.

We implement both weighting schemes, and compare our methods with Monte Carlo integration based methods under different sampling conditions. Our results show that both new schemes consistently provide better estimations for input testing functions. We discuss these two schemes and future work at the end of this paper.

2 Preliminaries

Consider a smooth orientable compact manifold M of dimension k that is isometrically embedded in some Euclidean space $\mathbb{R}^d$, and is equipped with a natural metric induced from the Euclidean metric. The *medial axis* of M is the closure of the set of points in $\mathbb{R}^d$ that have at least two closest points in M. For any $p \in M$, the *local feature size at* p, denoted by $\mathrm{lfs}(p)$, is the distance from p to the medial axis. The *reach* of M, denoted by ρ, is the infinum of the local feature size at any point in M. In this paper, we assume that the manifold M has a positive reach. We also assume that both k and d are known a priori. Note many algorithms have been developed to estimate the intrinsic dimension from point cloud with guarantees [9, 10, 12, 15].

Let P denote a set of sample points on M. We say that P is an *ε-sampling* of M if $p \in M$ for any $p \in P$, and for any point $x \in M$, there exists $q \in P$ such that $\|x - q\| \leq \varepsilon\rho$. P is an *(ε, δ)-sampling* of M if P is an *ε-sampling* of M and that any two points in P are at least $\delta\rho$ away from each other.

We use T_p to denote the tangent space of M at p, and $B(p, r)$ denotes the d-dimensional ball centered at p with radius r. In what follows, we use $\|x-y\|$ to denote the Euclidean distance between two points x and y, and $d_M(x, y)$ denote the geodesic distance between them on M for $x, y \in M$. For a point x and a set Y, the distance between them is defined as $d(x, Y) = \inf_{y\in Y} \|x - y\|$.

Integral estimation. Given a set of points $P = \{p_1, \ldots, p_n\}$, a function $f : M \to \mathbb{R}$ is represented as a vector $\mathbf{f} = [\mathbf{f}_1, \ldots, \mathbf{f}_n]$, where $\mathbf{f}_i = f(p_i)$ is its value at point p_i. We approximate the integral $\int_M f$ in a standard way by using a weighted sum: $\sum_i \omega_i \mathbf{f}_i$, which can be rewritten as the vector inner product $\langle \omega, \mathbf{f} \rangle$ for $\omega = \langle \omega_1, \ldots, \omega_n \rangle$. Hence developing an integral estimation simply means to construct a weighting scheme ω. We remark that one can consider the integral $\int_M f d\nu$ as the inner product between the function f and the volume measure $d\nu$. From this point of view, it is natural to approximate an integral as an inner product between vectors in the discrete setting, and the weighting vector ω should describe the volume measure at each sample point. This also implies that the weighting vector should be able to be constructed by local information, which we collect by constructing a *local patch* around each input point.

Local patches. Specifically, we build an approximation of the local manifold patch $M_\eta = M \cap B(p, \eta)$ for each $p \in P$ for some parameter $\eta \leq \rho/2$ as follows. Set $P_\eta = P \cap B(p, \eta)$.

(1) Construct from P_η a k-dimensional subspace $\tilde{\mathcal{T}}_p$ to approximate the tangent space T_p at p, using the algorithm in [7, 17].

(2) Project the set of points P_η onto $\tilde{\mathcal{T}}_p$.

(3) Let Π denote the projection from M_η onto $\tilde{\mathcal{T}}_p$. Build the Delaunay triangulation K_η of $\Pi(P_\eta)$ on $\tilde{\mathcal{T}}_p$.

K_η is a k-dimensional triangulation. The projection map $\Pi : M_\eta \to \tilde{\mathcal{T}}_p$ is injective for $\eta \leq \rho/2$. Let $\Phi : \Pi(M_\eta) \to M_\eta$ be its inverse. Below we cite some known results relating T_p, $\tilde{\mathcal{T}}_p$ and M [6, 7]. In particular, Lemma 2.1 (3) guarantees that the volume measure on K_η is indeed close to the measure on M_η as the Jacobian of this projection map is close to 1. Hence we can use K_η as a faithful approximation of M_η and estimate weights for integral from K_η.

Lemma 2.1 ([6, 7]) *Given a point cloud P that ε-samples M, for a fixed point p, perform the algorithm introduced above with parameter $10\varepsilon\rho \leq \eta \leq \rho/2$.*

(1) Given two points $p, q \in M$, let $d = \|p - q\| < \rho/2$. Then we have that $d \leq d_M(p, q) \leq d + O(d^3)$.

(2) The approximate tangent space $\tilde{\mathcal{T}}_p$ is close to T_p and the angle between them is $\angle(T_p, \tilde{\mathcal{T}}_p) = O(\eta/\rho)$.

(3) The Jacobian of the map Π and its inverse Φ at any point $x \in M_\eta$ are bounded respectively by:

$$1 - O\left(\frac{\|x - p\|^2}{\rho^2} + \frac{\eta}{\rho}\right) \leq J(\Pi)|_x \leq 1, \quad \text{and}$$
$$1 \leq J(\Phi)|_{\Pi(x)} \leq 1 + O\left(\frac{\|x - p\|^2}{\rho^2} + \frac{\eta}{\rho}\right).$$

Furthermore, the smallest eigenvalue of $J(\Pi)|_x$ is lower-bounded by $1 - O(\|x - p\|^2/\rho^2 + \eta/\rho)$, while the largest eigenvalue of $J(\Phi)|_{\Pi(x)}$ is upper-bounded by $1 + O(\|x - p\|^2/\rho^2 + \eta/\rho)$.

3 Weighting Schemes

In this section, we describe two somewhat dual weighting schemes for integral estimation: the Voronoi weighting scheme and the Principal Eigenvector weighting scheme.

3.1 Voronoi Weighting Scheme

In this section, we choose and fix the parameter $\eta = c\varepsilon\rho$ for some constant $c > 10$. For each point $p \in P$, we construct the local patch and its triangulation K_η as described before, and consider the dual Voronoi diagram of K_η in $\tilde{\mathcal{T}}_p$. Let $\tilde{\mathcal{V}}(p)$ denote the Voronoi cell of p in $\tilde{\mathcal{T}}_p$. Take the area of $\tilde{\mathcal{V}}(p)$ as the weight of p, denoted by $\tilde{\mathcal{A}}_p$. The *Voronoi* weighting scheme is simply setting $\omega_i = \tilde{\mathcal{A}}_{p_i}$, and we compute the integral as $\langle \omega, \mathbf{f} \rangle = \sum_i \tilde{\mathcal{A}}_{p_i} f(p_i)$.

Now let $Vor(P)$ be the geodesic Voronoi diagram of P on M. Let $V(p)$ denote the Voronoi cell of p in $Vor(P)$ and A_p its area. Intuitively, A_p serves as a good weighting scale for the sample point p and can be used to approximate the integral. Hence to show the theoretical guarantee of our weighting scheme, the goal is to bound the relation between A_p (which cannot be computed) with the area $\tilde{\mathcal{A}}_p$ in the approximate tangent space $\tilde{\mathcal{T}}_p$. Note that the Voronoi neighbors of p on M and those in $\tilde{\mathcal{T}}_p$ are not necessarily the same, and one can in fact construct examples where they induce arbitrarily different areas for some ε-sampled point sets. Hence we need to consider a sampling condition which is more restricted than the ε-sampling condition — specifically, we assume that P is an (ε, δ)-sampling of M, and the convergence result will be achieved as long as $\delta = \Omega(\varepsilon^{3/2-\xi})$ for any $\xi > 0$ (the smaller δ is, the more general the sampling condition is). Recall that we have chosen $\eta = c\varepsilon\rho$.

Lemma 3.1 *Assume P is an (ε, δ)-sampling of M with sufficiently small $\varepsilon < 1/20$. Consider the point $p \in P$ and any point $q \in P_\eta = P \cap B(p, \eta)$. Let $\tilde{S}$ be the bisector between p and $\Pi(q)$ on $\tilde{\mathcal{T}}_p$. For any $x \in M_\eta = M \cap B(p, \eta)$ with $d_M(x, p) = d_M(x, q)$, we have that $d(\Pi(x), \tilde{S}) = O(\varepsilon^3 \rho/\delta)$.*

Proof: Denote $\tilde{y} = \Pi(y)$ for any $y \in M_\eta$ (note that $\tilde{p} = p$). Now take an arbitrary point $x \in M_\eta$ that is on the geodesic bisector between p and q on M. Set $L = d_M(x, p) = d_M(x, q)$. Since $M_\eta = M \bigcap B(p, \eta)$, $L = O(\eta) = O(\varepsilon\rho)$ by Lemma 2.1 (1). First, we show that $\|p - \tilde{x}\| = L(1 - c_1\varepsilon)$ and $\|\tilde{q} - \tilde{x}\| = L(1 - c_2\varepsilon)$ for some constants c_1, c_2. We show the inequality for $\|p - \tilde{x}\|$. That for $\|\tilde{q} - \tilde{x}\|$ is symmetric.

Specifically, let $\gamma(x_1, x_2)$ denote a minimizing geodesic path between x_1 and $x_2 \in M$. Now, for any point $y \in \gamma(x, p)$ or $y \in \gamma(x, q)$, it follows from $L = O(\varepsilon\rho)$ and Lemma 2.1 (1) that $\|y - p\| = O(\varepsilon\rho)$. Consider the projection of $\gamma(x, p)$ onto $\tilde{\mathcal{T}}_p$ through the projection map Π, and let $len(\Pi(\gamma(x, p)))$ denote its length. We have that :

$$\lambda_{min} \cdot d_M(x, p) \;\leq\; len(\Pi(\gamma(x, p))) \;\leq\; d_M(x, p) \;=\; L,$$

where λ_{min} is the smallest eigenvalue of the Jacobian of Π at any point $y \in \gamma(x, p)$. It then follows from Lemma 2.1 (3) and $\|y - p\| = O(\varepsilon\rho)$ for $y \in \gamma(x, p)$, that $\lambda_{min} = 1 - O(\varepsilon)$, implying that

$$len(\Pi(\gamma(x, p))) \;=\; (1 - a\varepsilon)L$$

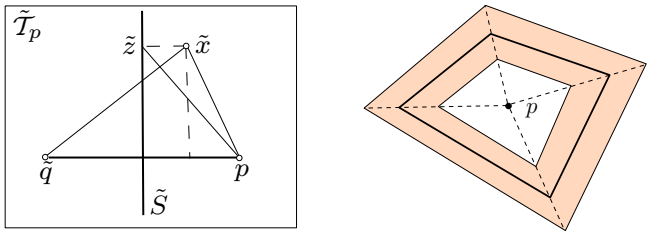
for some constant a. Since $\|\tilde{x} - p\| \leq len(\Pi(\gamma(x, p)))$, it then follows that $\|\tilde{x} - p\| \leq (1 - a\varepsilon)L$.

On the other hand, consider the image of the line segment $\tilde{x}p$ under the inverse map $\Phi = \Pi^{-1}$. Let L' denote the length of this curve on M. It is bounded by the following inequality:

$$L = d_M(x, p) \;\leq\; L' \;\leq\; \lambda_{max} \cdot \|\tilde{x} - p\|,$$

where λ_{max} is the largest eigenvalue of the Jacobian of Φ at any point y in segment $\tilde{x}p$. Hence the length of L' is $\|\tilde{x} - p\|(1 + b\varepsilon)$ for some constant b by Lemma 2.1(3), which implies that

$$\|\tilde{x} - p\| \;\geq\; L/(1 + b\varepsilon).$$



(a) (b)

Figure 1: Illustrations for the proofs of Lemma 3.1 in (a) and of Lemma 3.2 in (b).

Combining this with the earlier inequality $\|\tilde{x} - p\| \leq (1 - a\varepsilon)L$, we have that $\|\tilde{x} - p\| = (1 - c_1\varepsilon)L$ for some constant c_1. By a similar argument, we can show $\|\tilde{q} - \tilde{x}\| = (1 - c_2\varepsilon)L$ for some constant c_2.

We now show that $d(\tilde{x}, \tilde{S}) = O(\varepsilon^3\rho/\delta)$. Let $\tilde{z}$ be the projection of $\tilde{x}$ in the bisector $\tilde{S}$, which is a $(k-1)$-dimensional hyperplane. Note that $\tilde{x}\tilde{z}$ is parallel to $p\tilde{q}$ in the k-dimensional space $\tilde{\mathcal{T}}_p$. See Figure 1 (a) for an illustration — by elementary calculations, we have that:

$$\begin{aligned} & \|\tilde{x} - \tilde{q}\|^2 - \|\tilde{x} - \tilde{p}\|^2 = 2\|p - \tilde{q}\| \cdot \|\tilde{x} - \tilde{z}\| \\ \Rightarrow \quad & d(\tilde{x}, \tilde{S}) = \|\tilde{x} - \tilde{z}\| = \frac{\|\tilde{x} - \tilde{q}\|^2 - \|\tilde{x} - \tilde{p}\|^2}{2\|p - \tilde{q}\|} \ . \end{aligned}$$

It then follows from earlier results that

$$d(\tilde{x}, \tilde{S}) = O(L^2\varepsilon)/\|p - \tilde{q}\| = O(\frac{\varepsilon^3\rho}{\delta}),$$

where the second equality uses the fact that P is δ-sparse and thus $\|p - \tilde{q}\| = \Omega(\delta\rho)$. ■

Lemma 3.2 *Given an (ε, δ)-sampling P of M with sufficiently small ε, and $\delta = \Omega(\varepsilon^{3/2})$, for any $p \in P$, we have:*

$$(1 - O(\varepsilon + \frac{\varepsilon^3}{\delta^2}))A_p \ \leq \ \tilde{\mathcal{A}}_p \ \leq \ (1 + O(\varepsilon + \frac{\varepsilon^3}{\delta^2}))A_p.$$

Proof: Let $Q \subset P$ be the set of geodesic Voronoi neighbors of p in $Vor(P)$ on M and $R \subset P$ the set of points such that $\Pi(R)$ is the set of Voronoi neighbors of p in $\tilde{\mathcal{T}}_p$. Since P is an (ε, δ)-sampling of M and the projection map Π has a Jacobian close to 1, by choosing $\eta = c\varepsilon\rho$ for some c big enough, we have that both $Q, R \subseteq P_\eta = P \cap M_\eta$. Hence $\Pi(R)$ is also the set of neighbors of p in the dual Voronoi diagram of K_η. Due to the sampling conditions of P, observe that $\|x - p\| = O(\varepsilon\rho)$ for any $x \in V(p)$, or $x \in \tilde{\mathcal{V}}(p)$, and $\|y - p\| = \Omega(\delta\rho)$ for any $y \in \partial V(p)$ or $y \in \partial\tilde{\mathcal{V}}(p)$.

For each $w \in P$, let $\tilde{S}_w$ be the bisector between p and $\tilde{w}$ on $\tilde{\mathcal{T}}_p$, which is a $(k-1)$-dimensional hyperplane. Now set $\beta = \frac{c'\varepsilon^3}{\delta^2}$ for some constant c' big enough. Let $\tilde{S}_w^+ \subset \tilde{\mathcal{T}}_p$ be the hyperplane parallel to $\tilde{S}_w$ but further away from p by $\beta\|\tilde{w} - p\|$, and $\tilde{S}_w^-$ the hyperplane parallel to $\tilde{S}_w$ but closer to p by $\beta\|\tilde{w} - p\|$. Let H_w^+ and H_w^- be the halfspaces in $\tilde{\mathcal{T}}_p$ containing p bounded by $\tilde{S}_w^+$ and $\tilde{S}_w^-$ respectively. It is important to note that $\cap_{w \in P_\eta} H_w^- = \cap_{w \in R} H_w^-$ and $\cap_{w \in P_\eta} H_w^+ = \cap_{w \in R} H_w^+$, as every hyperplane is moved via the same ratio w.r.t its distance to p. Hence we have

$$(\cap_{w \in R} H_w^-) \ \subset \ \tilde{\mathcal{V}}(p) \ \subset \ (\cap_{w \in R} H_w^+). \tag{1}$$

See Figure 1 (b) for an illustration, where the thick polygon is the boundary of $\tilde{\mathcal{V}}(p)$, and the shaded region is

$$(\cap_{w\in R} H_w^+) \ \setminus \ (\cap_{w\in R} H_w^-).$$

The volume of $(\cap_{w\in R} H_w^-)$ can be lower bounded by

$$\tilde{\mathcal{A}}(p)\ (1-k\beta) = \tilde{\mathcal{A}}(p)\ \big(1 - O(\frac{\varepsilon^3}{\delta^2})\big).$$

Similarly the volume of the outer convex region $(\cap_{w\in R} H_w^+)$ can be upper bounded by $\tilde{\mathcal{A}}(p)(1 + O(\frac{\varepsilon^3}{\delta^2}))$.

Next, we claim the following, which implies that $\tilde{\mathcal{V}}(p)$ and $\Pi(V(p))$ are sandwiched within the same region.

$$(\cap_{w\in R} H_w^-) \ \subset \ \Pi(V(p)) \ \subset \ (\cap_{w\in R} H_w^+). \tag{2}$$

Indeed, for any $w \in P_\eta$, note that the distance between H_w^- or H_w^+ to H_w is $\beta\|\tilde{w} - p\|$, which is lower-bounded by $\Omega(\frac{\varepsilon^3\rho}{\delta})$ as the input is an (ε, δ)-sample. Hence by Lemma 3.1, each geodesic bisector, say the one between w and p, is projected to the slab between H_w^- and H_w^+. Since the map Π is locally homeomorphic, one can show that the projection of the boundary of $V(p)$ is contained within the sandwich region $\cap_{w\in P_\eta} H_w^+ - \cap_{w\in P_\eta} H_w^-$ which is the same as $\cap_{w\in R} H_w^+ - \cap_{w\in R} H_w^-$. Eqn(2) thus holds. Combined with the bounds on the volume of these intersections, we have

$$\tilde{\mathcal{A}}_p(1 - \frac{\varepsilon^3}{\delta^2}) \le Area(\Pi(V(p))) \le \tilde{\mathcal{A}}_p(1 + \frac{\varepsilon^3}{\delta^2}). \tag{3}$$

Furthermore, since $\|x - p\| = O(\varepsilon\rho)$ for any $x \in V(p)$, by Lemma 2.1 (3) we can bound the area $A_p = Area(V(p))$ by

$$Area(\Pi(V(p))) \ \le \ A_p \ \le \ \big(1 + O(\varepsilon)\big) Area(\Pi(V(p))). \tag{4}$$

The lemma then follows from Eqns (3, 4). ■

Theorem 3.3 *Given an (ε, δ)-sampling P of M with ε sufficiently small, compute $\tilde{\mathcal{A}}_p$ for each $p \in P$ as described above. Then for any Lipschitz function f we have that*

$$\left| \int_M f - \sum_{p\in P} \tilde{\mathcal{A}}_p f(p) \right| = \ O(\varepsilon + \varepsilon^3/\delta^2),$$

implying that for $\delta = \Omega(\varepsilon^{3/2-\xi})$ with any positive constant ξ, we have

$$\lim_{\varepsilon\to 0} \left| \int_M f - \sum_{p\in P} \tilde{\mathcal{A}}_p f(p) \right| = \ 0.$$

Proof: Let L be the Lipschitz constant of f. Note that the geodesic Voronoi cells of points in P form a partition of the manifold M. For any point $x \in M$, let $N_P(x)$ denote its nearest neighbor in P. Since $d_M(x, N_P(x)) = O(\varepsilon\rho)$ for any $x \in M$, we have that

$$\begin{aligned} \left| \int_M f dx - \sum_{p\in P} f(p) A_p \right| &= \left| \int_M [f(x) - f(N_P(x))] dx \right| \\ &\le \int_M |f(x) - f(N_P(x))| \, dx \\ &= O(\varepsilon\rho \cdot L \cdot vol(M)) = O(\varepsilon), \end{aligned} \tag{5}$$

where $vol(M)$ is the volume of the manifold M. On the other hand, by Lemma 3.2, we have that

$$\left|\sum_{p\in P} f(p)A_p - \sum_{p\in P} f(p)\tilde{\mathcal{A}}_p\right| = \|f\|_\infty \sum_{p\in P}(A_p O(\varepsilon + \varepsilon^3/\delta^2)) = O(\varepsilon + \varepsilon^3/\delta^2). \quad (6)$$

The theorem then follows from Eqns (5, 6), and the big-O notation hides constants both related to the underlying manifold M and to the input Lipschitz function f. ∎

Remark. Our theoretical guarantee requires that the algorithm knows $\varepsilon\rho$, and thus can choose η appropriately. Such information is typically hard to estimate in practice. In our implementation, we simply choose η as a constant times the average distance between a sample point and its nearest neighbor in P.

3.2 Principal Eigenvector Weighting Scheme

It is well-known that the integral of any function defined on a smooth compact manifold M is preserved under the heat diffusion process. In other words, let $\mathcal{H}_t$ denote the heat operator w.r.t. time t. For any $f : M \to \mathbb{R}$ and any t, we have that $\int_M f(x)dx = \int_M \mathcal{H}_t f(x)dx$. We call this the *heat-preservation property*. This is a fundamental property of the heat operator, and a useful one in practice. For example, heat diffusion has been widely used to smooth functions on a manifold. The heat-preservation property means that the function will not keep decreasing to zero during the smoothing process. Now, given an ε-sampling P of M, we wish to develop a discrete integral estimation scheme so that certain discrete heat diffusion is an integral invariant.

First, suppose we are given a discrete heat operator H_t, analogous to $\mathcal{H}_t$ in the smooth case. Being a linear operator on functions, H_t can be represented as an $n \times n$ matrix, where n is the number of points in P. The heat-preservation property means that for any $\mathbf{f}$,

$$< \omega, \mathbf{f} > \;=\; < \omega, \mathrm{H}_t\mathbf{f} > \quad \Rightarrow \quad \omega^T\mathbf{f} = \omega^T\mathrm{H}_t\mathbf{f}.$$

Hence $\omega^T = \omega^T\mathrm{H}_t$, implying that ω is *necessarily* a left-eigenvector of H_t with eigenvalue being 1. Hence we now need a good discrete heat operator, which indeed has a left-eigenvector corresponding to eigenvalue 1.

Constructing discrete heat operator. To this end, we construct H_t using a similar algorithm as the one proposed in [7] to approximate the so-called Laplace-Beltrami operator Δ, which is connected to the heat operator by the standard relation $H_t = e^{-t\Delta}$ [22]. Specifically, first, we modify the construction of the local patch (presented in Section 2) around each sample point p_i slightly: In Step (1), we approximate the tangent space at p_i using sample points that are within the ball $B(p_i, 10\varepsilon\rho)$ around p_i. In Step (2) and (3), we use a different neighborhood size η which is a small constant. We then construct the Delaunay triangulation $K_\eta(p_i)$ for the projection of points from $P_\eta = P \cap B(p_i, \eta)$ in the approximate tangent space at p_i. Let $P_{\eta/2}(p_i)$ denote the set of sample points within the ball $B(p_i, \eta/2)$. We now define a weighting value A_i^j as follows:

$$A_j^i = \begin{cases} \frac{1}{k+1}\sum_\sigma Area(\sigma) & \text{if } p_j \in P_{\eta/2}(p_i) \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

and the summation is taken over all k-simplices σ incident to $\Pi(p_j)$ in the local patch $K_\eta(p_i)$ at p_i.

Finally, we fix the "time" parameter $t = \Omega(\frac{1}{\varepsilon^{2+\xi}})$ for any constant $\xi > 0$. We now construct the discrete Heat operator H_t, which is also an $n \times n$ matrix, by setting the jth element in the ith row as:

$$\mathrm{H}_t[i,j] = \frac{G_{ij} A_i^j}{\sum_{k=1}^{n} G_{ik} A_i^k}, \quad \text{where} \quad G_{ij} = e^{-\frac{\|p_i - p_j\|^2}{4t}}.$$

In other words, $\mathrm{H}_t f(p_i)$ is a convex combination of all $f(p_j)$s where p_j is within distance $\eta/2$ to p_i. The weight of each $f(p_j)$ is based on its area in the local triangulation $K_\eta(p_i)$ and the distance from p_j to p_i.

Properties of H_t. Intuitively, the above construction follows the idea first proposed in [4] to use the Gaussian kernel $G_t(x,y) = \frac{1}{(4\pi t)^{k/2}} e^{-\frac{\|x-y\|^2}{4t}}$ to approximate the so-called heat kernel $h_t(x,y)$. It also relates to the discrete Laplace operator L_t constructed in [7] by $\mathrm{H}_t = I - tD^{-1}L_t$, where D is a diagonal matrix with

$$D[i][i] = \frac{1}{(4\pi t)^{k/2}} \sum_{k=1}^{n} G_{ik} A_i^k.$$

This intuitively approximates the first order Taylor expansion of $H_t = e^{-t\Delta}$. We can show that $\|\mathrm{H}_t f - \mathcal{H}_t f\|_\infty = O(t^\alpha)$ for any Lipschitz function f, where α is some positive constant. Hence for small t, the discrete heat operator is close to the heat operator. The proof is straightforward but quite technical, following from several results and procedures in [4, 5]. We thus omit it from this extended abstract.

In general, since H_t is not symmetric, it may have complex eigenvalues and eigenvectors. In our case, however, since the row sum of H_t is 1, and all elements in the matrix are non-negative, it is easy to check that the constant vector $[1, 1, \ldots, 1]^T$ is a *right*-eigenvector of H_t corresponding to eigenvalue 1. Furthermore, since H_t is an averaging operator, the largest possible eigenvalue of H_t is 1. Hence we obtain the following.

Lemma 3.4 *The maximum eigenvalue of* H_t *is* 1*, and its corresponding right and left eigenvectors are real-valued.*

Principal Eigenvector weighting scheme. Since the left-eigenvector μ of H_t with eigenvalue 1 is real valued, we can now set the weighting vector $\omega = \mu$. Obviously, $\langle \omega, \mathbf{f} \rangle = \langle \omega, \mathrm{H}_t \mathbf{f} \rangle$ for any $\mathbf{f}$. Unfortunately, it is not clear whether one can bound the relation between $\langle \omega, \mathbf{f} \rangle$ and $\int_M f(x)dx$. However, if we can compute a global mesh from input point cloud that approximates the underlying manifold M in same way as defined in [6], and use the global mesh to set up A_i^j instead of using local patches, then, it can be shown that

$$\lim_{t \to 0} \left| \langle \omega, \mathbf{f} \rangle - \frac{1}{vol(M)} \int_M f(x)dx \right| = 0,$$

where $vol(M)$ represents the volume of the manifold M.

Remark 1. We remark that since the eigenvector constructed is normalized, we can only estimate the integral up to a scaling factor, which is the volume of the manifold M. This limitation also exists for Monte-Carlo integration based methods. The Voronoi weighting scheme approximates the integral without any scaling factor.

Remark 2. Note that each row of our discrete Heat operator H_t is computed based on the same local patch information as in the case for Voronoi weighting scheme. However, Heat operator

H_t averages over a much bigger neighborhoods, not just one Voronoi cell, which may make it not sensitive to the δ-sparsity sampling condition. In fact, this is observed in the experments, see Section 4. In addition, taking the left-eigenvector of this matrix in some sense connects across different local patches. Such a global connection is intuitively necessary as we wish to guarantee the heat-preservation property, which is a global behavior. On the other hand, in the Voronoi weighting scheme, we only wish to locally approximate the metric on the manifold. Hence no transformation of information across local patches is needed.

4 Experiments

In this section, we implement the two weighting schemes as described in Section 3 and compare them with Monte Carlo integration. In Monte Carlo integration, we use kernel density estimation (KDE) Toolbox developed by Ihler and Mandel [18] to obtain the weights for input sample points and estimate the integral as weighted sum of the function values at sample points. We also show the integral estimation results using equal weights at every sample point. The experiment results show that our schemes consistently outperform Monte Carlo integration in estimation accuracy, especially when there are noise in either the integrand function or in input sample points.

Experimental setup. We consider two data sets. One is a synthetic data set where samples are drawn from a flat 2-torus T^2 which is defined parametrically as

$$T^2(\alpha, \beta) = (\cos\alpha, \sin\alpha, \cos\beta, \sin\beta)$$

with both α and β ranging from 0 to 2π. The need of such synthetic data is so that the ground truth can be computed. We have also experimented with other synthetic data, such as 2-sphere and 3-torus. The results are similar and thus we only focus on T^2 here. We perform our experiments on three types of sampling conditions of input points. To obtain a uniform-sampling of T^2, we simply uniformly sample the parameter domain (a square) since the parametrization is isometric. To achieve a non-uniform sampling of T^2, we impose a distribution of Gaussian mixture on the parameter domain and accept a sample point with higher probability if the Gaussian mixture has a bigger value at the sample point. We also experiment with a skew-sampled point cloud of the underlying manifold that violates the δ-sparsity condition. Specifically, in the skew-sampled input, half of the points are drawn from a very narrow vertical slab in the middle of the parameter domain. Finally, we also experiment with point cloud data with noise, which is produced by perturbing the coordinates of input points in the ambient space.

The second data set is a real molecular model as shown in the Figure 2. There are around 8000 sample points (shown as blue dots) on the surface. Since we do not know the underlying manifold, we use the underlying mesh of this model to obtain a "ground truth" — Given the function values at the vertices, we linearly interpolate the function within each mesh facet and take the integral of this piecewise linear function over the mesh as the ground truth. It is shown in [6] that this mesh-integral is close to the true integral.

We use two groups of test functions in our experiments: trigonometric functions ($sinx$, $cosx$, $sin2x$, $cos2x$, $sin3x$, $cos3x$, $sin4x$, $cos4x$) and polynomial functions (x, x^2, x^3). Appropriate constants are added to test functions to guarantee that the integral is not close to 0. We also have noisy versions of these functions by perturbing the values at each sample point randomly.

Note that except for the Voronoi weighting scheme, other methods generate weights up to a scaling factor, which is the manifold volume. Hence to compare different schemes, we normalize the weights so that $\sum_i w_i$ equals the area of the underlying manifold. To measure the error with

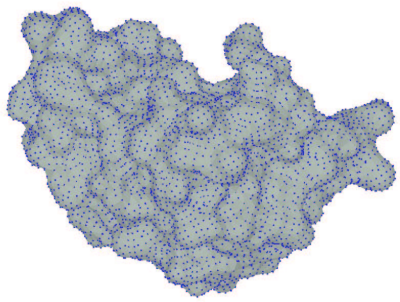


Figure 2: The molecular surface of the protein (PDB id: 1BRS, chain A) with sample points.

the ground truth I, we use relative error $|I - I_D|/I$, where I_D is the discrete approximation. Due to the space limitation, we average all the relative errors from the same group of basis functions and report that as the estimation error of a particular method.

Accuracy and convergence. Table 1 and Table 2 show the results for the synthetic data T^2, embedded in the 4-dimensional Euclidean space. Table 1 includes presents results for the uniform and non-uniform sampling of T^2, while Table 2 shows those for the skew-sampling point cloud data. The KDE toolkit provides five variants for density estimation accouting for five different ways to estimate the width of the kernel Gaussian function. Due to space limitation, in each case, we report the two KDE variants that produce the most accurate estimations. In the case of the uniform and non-uniform sampled data, we present results from the *rule of thumb (rot)* and the *maximum smoothing principle (msp)* KDE methods. In the case of skew-sampled data, we present results from the KDE (rot) and the more sophisticated *likelihood cross-validation (lcv)* methods.

Table 1 shows that both of our schemes show convergent estimation for the uniform sampled and nonuniform sampled point cloud data, as more sample points are drawn from the underlying manifold, while all methods of Monte Carlo integration do not yet. It is interesting to note that Monte Carlo integration is known to be statistical convergent, and our input point cloud data are in fact drawn from Gaussian distribution in these cases. Table 1 perhaps suggests that the convergent rate for Monte Carlo integration is much slower than our schemes in experiments. Our methods also achieve better estimation in general than KDE based methods.

For the skew-sampled data, Table 2 shows that our methods still show convergent behavior. Specifically, the Voronoi weighting scheme seems to be both accurate and convergent, even though there is no theoretical guanratee for such inputs (where the δ-sparsity condition is not met).

We also remark that as the dimension of the ambient space increases, our methods return exactly same results, while the performances of KDE-based methods further deteriorate. This is partly due to the fact that our algorithm assumes that the input dimension is given. However, it is unclear how to use this information for KDE-based methods.

Table 4 shows the results for the molecular data, and similar behavior can be observed. In this case, we present the results by KDE (rot) and KDE (msp) methods, which produce best numerical estimations for this data.

Timing. Table 4 compares the timing of different methods. We note that the two simplest KDE methods (KDE rot and KDE msp) are very efficient. The more complicated KDE (lcv) method runs slower than the Voronoi weighting scheme. The Principal Eigenvector scheme is slowest, as

	Voronoi	Eigen	KDE(rot)	KDE(msp)	Equal
uniform	4 /<1/<1	21 / 9 / 1	25 / 12 / 11	19 / 11 / 9	29 / 59 / 35
nonuniform	3 / 1 /<1	5 / 5 / 3	22 / 4 / 11	25 / 4 / 11	108/ 86 / 64
uniform with noisy vert.	5 / 3 / 2	16 / 1 / 1	26 / 13 / 12	17 / 12 / 10	29 / 59 / 35
nonuniform with noisy vert.	1 / 3 / 1	3 / 3 / 2	23 / 5 / 13	27 / 5 / 12	108/ 86 / 64
uniform with noisy fun.	33 / 23 / 24	27 /17/22	54 / 35 / 22	44 / 33 / 19	14 / 50 / 45
nonuniform with noisy fun.	21 / 13 / 12	16 /16/13	9 / 14 / 20	12 / 13 / 21	105/ 91 / 68
errors from polynomial functions					
uniform	14 / 7 / 2	22 /15/ 6	116/ 67 / 95	143/ 81 /102	474/331/342
nonuniform	21 / 27 / 7	64 /52/18	140/302/223	174/334/245	403/677/599
uniform with noisy vert.	20 / 22 / 19	87 /29/15	124/ 74 / 89	151/ 85 / 97	474/331/342
nonuniform with noisy vert.	11 / 23 / 20	53 /43/28	128/295/213	164/328/237	403/677/599
uniform with noisy fun.	167/ 95 / 60	182/93/53	150/105/153	195/117/160	534/358/373
nonuniform with noisy fun.	211/ 91 / 46	237/94/62	226/291/248	256/322/270	495/641/613
errors from trigonometric functions					

Table 1: Relative error of integral estimated by various methods for 2500/5000/10000 points from flat-2 torus T^2 embedded in $\mathbb{R}^4$. The last three columns are results based on the Monte Carlo integration, by using density estimation methods provided by the KDE tool, and by equal weight, respectively. All numbers are scaled by 10^{-4}.

	Voronoi	Eigen	KDE(rot)	KDE(lcv)	Equal
skew-sample	10/ 8 / 3	73 / 14 / 5	653 / 514 / 647	610 / 706 / 668	5742 / 5747 / 5797
skew-sample (noisy vert.)	12/ 8 /14	115/ 68 /135	645 / 516 / 621	665 / 755 / 773	5742 / 5747 / 5797
skew-sample (noisy fun.)	21/22/ 8	85 / 26 / 11	642 / 510 / 645	609 / 711 / 660	5737 / 5753 / 5792
errors from polynomial functions					
skew-sample	33/20/13	316/110/ 42	8702/9898/10407	2559/2975/2866	23300/23624/23619
skew-sample (noisy vert.)	31/64/45	378/246/461	8507/9524/ 9937	2692/3088/3246	23300/23624/23619
skew-sample (noisy fun.)	77/57/31	326/ 98 / 43	8731/9906/10402	2586/2962/2868	23312/23621/23617
errors from trigonometric functions					

Table 2: Relative error of integral estimated by various methods for 2500/5000/10000 skew-sampled points from flat-2 torus T^2 embedded in $\mathbb{R}^4$. All numbers are scaled by 10^{-4}.

	Voronoi	Eigen	KDE(rot)	KDE(msp)	Equal
original	8	50	79	103	221
noisy vert.	8	60	80	103	221
noisy fun.	43	43	98	122	202
errors from polynomial functions					
original	58	199	452	465	366
noisy vert.	119	179	462	470	366
noisy fun.	219	190	432	439	425
errors from trigonometric functions					

Table 3: Relative error of integral estimated by various methods for molecular data. All numbers are scaled by 10^{-4}.

Voronoi	Eigen	KDE(rot)	KDE(msp)	KDE(lcv)	Equal
7/18/51	38/167/734	¡1/2/ 6	¡1/2/ 7	21/60/210	¡1/¡1/¡1

Table 4: Timing in seconds of various methods for 2500/5000/10000 points from flat-2 torus T^2 embedded in $\mathbb{R}^4$.

method	4d	20d	40d	60d
Voronoi	7 / 18 / 51	10/ 28 /100	13/ 40 /146	16/ 53 /199
Eigen	34/167/734	41/174/772	44/190/824	47/199/883

Table 5: Timing in seconds for 2500/5000/10000 points from flat-2 torus T^2 embedded in $\mathbb{R}^4$, $\mathbb{R}^{20}$, $\mathbb{R}^{40}$ and $\mathbb{R}^{60}$.

it requires to first construct the discrete Laplace operator. We point out that the timing of our methods is based on a rather straightforward implementation, without any optimization. Table 4 shows that as expected, the ambient dimension has only mild effect on the running time of our proposed methods. (Note that we assume that the intrinsic dimension is known.)

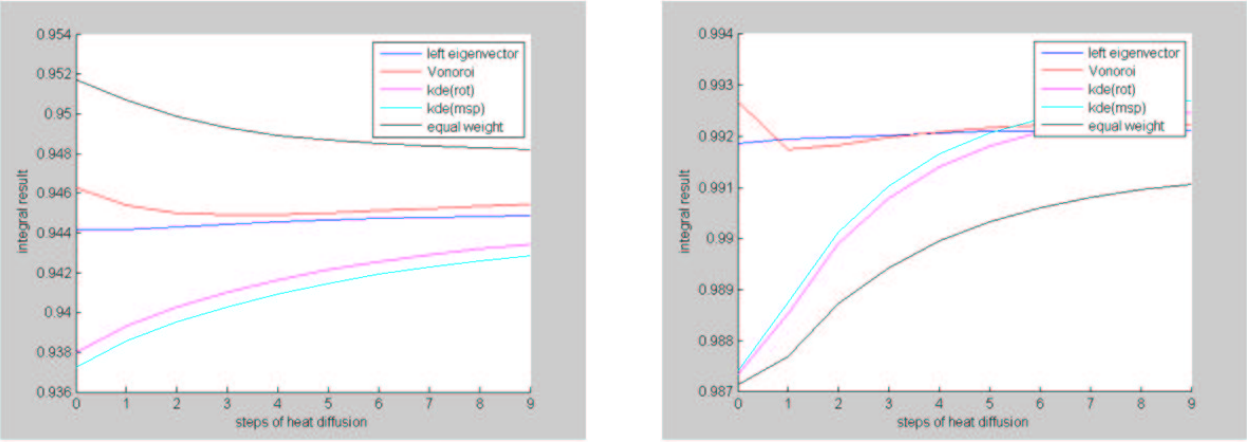
Stability. To test the stability of these methods under heat diffusion smoothing, we use heat diffusion operator (as constructed by our algorithm) to smooth a function for a few times. At each iteration, we compute the integral and see how the results change. Figure 3 shows the trend for two test functions. The experiments are conducted on the molecular data. By definition, the integral under the Principal Eigenvector scheme is invariant; while obviously, no other method preserves integrals during this smoothing.

5 Conclusion and Discussion

In this paper, we considered the problem of estimating the integral of a function over a submanifold embedded in the high dimensional space from a set of sample points. We approached the problem from a geometric point of view and developed two schemes based on the local geometric approximation of the manifold. Both schemes have a running time depending only linearly on the dimension of the ambient space. We showed that the Voronoi scheme converges to the ground truth under mild assumptions on input sample points and the integrand function. The Principal Eigenvector scheme is constructed w.r.t a discrete Heat operator, and preserves heat under the diffusion process as guided by that operator. These are the first results of this sort for integral estimation from general PCD. In our experiments, both new schemes give convergent results, and produce numerically much better estimations than the standard Monte Carlo integration methods. The Voronoi scheme usually produces more accurate results, while the Principal Eigenvector scheme has the additional property that it has an accompanying Heat operator which is an integral invariant under this scheme.

Our Voronoi scheme still requires that input points (ε, δ)-sample the underlying manifold M. A natural question is whether it is possible to remove the δ-sparsity constraint. This seems to require a more global approach to estimate error than our current local method used in Section 3.1.

Our experiments show that the Voronoi scheme provides more accurate integral estimations than statistical methods even for data uniformly sampled. This intuitively is because that the former takes advantage of the local geometry (via the local patches) to estimate a more accurate



(a) (b)

Figure 3: (a) Estimated Integral for function $f = sinx + 1$ under heat diffusion. (b) Estimated Integral for function $f = cosx + 1$ under heat diffusion.

volume measure around each sample point. This raises an interesting question: how can one use local geometry to augment traditional statistical methods to achieve more accurate numerical estimations even under the statistical setting.

We believe that ideas behind the Principal Eigenvector scheme presents an interesting step towards obtaining global behavior from local, inconsistent patches. Sometimes it is enough to approximate the local measure on the manifold. However, many times certain stitching across local patches is necessary. The general question is how to achieve that without going to the extreme of constructing a global mesh. In our case, the global connection is made by taking the left eigenvector of the Heat operator. It will be interesting to quantify precisely what this left eigenvector captures. We conjecture that it in fact encodes a consistent set of area weights for sample points, and the Principal Eigenvector scheme converges to the true integral.

Finally, we remark that the Principal Eigenvector scheme can be considered as a by-product of computing the Laplace / Heat operator. All three are based on the same framework to discretize differential geometry information from point clouds data. It will be interesting to explore what other differential geometry quantities can be discretized within the same framework. Specifically, given that the Laplace operator encodes rich intrinsic geometry information about the underlying manifold, the goal is to construct the Laplace operator as a pre-processing step, and then retrieve a family of quantities from this fundamental operator.

Acknowledgement. The authors wish to thank Mikhail Belkin for many helpful discussions, and anonymous reviewers for very useful feedbacks. This work was supported in part by the Department of Energy (DOE) under grant DE-FG02-06ER25735, by the National Science Foundation (NSF) under grants CCF-0747082 and DBI-0750891, and by DARPA under grant HR0011-05-1-0007.

References

[1] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discrete Comput. Geom.*, 22:481–504, 1999.

[2] N. Amenta, S. Choi, T. K. Dey, and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. *Internat. J. Comput. Geom. Appl.*, 12:125–141, 2002.

[3] M. Belkin and P. Niyogi. Laplacian Eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[4] M. Belkin and P. Niyogi. Towards a theoretical foundation for laplacian-based manifold methods. In *Proc. 18th Annu. Comp. Learn. Theo.*, pages 486–500, 2005.

[5] M. Belkin and P. Niyogi. Convergence of Laplacian Eigenmaps. Preprint, 2008.

[6] M. Belkin, J. Sun, and Y. Wang. Discrete Laplace operator on meshed surfaces. In *Proc. 24st Annu. ACM Sympos. Comput. Geom.*, pages 278–287, New York, NY, USA, 2008. ACM.

[7] M. Belkin, J. Sun, and Y. Wang. Constructing Laplace operator from point clouds in $\mathbb{R}^d$. In *Proc. 20th ACM-SIAM Sympos. Discrete Algorithms*, pages 1031–1040, 2009.

[8] J. D. Boissonnat, L. J. Guibas, and S. Y. Oudot. Manifold reconstruction in arbitrary dimensions using witness complexes. In *Proc. 23st Annu. ACM Sympos. Comput. Geom.*, pages 194–203, 2007.

[9] S.-W. Cheng and M.-K. Chiu. Dimension detection via slivers. In *Proc. 20th ACM-SIAM Sympos. Discrete Algorithms*, pages 1001–1010, 2009.

[10] S.-W. Cheng, Y. Wang, and Z. Wu. Provable dimension detection using principal component analysis. *Internat. J. Comput. Geom. Appl.*, 18:414–440, 2008.

[11] T. K. Dey. *Curve and Surface Reconstruction: Algorithms with Mathematical Analysis*. Cambridge University Press, 2007.

[12] T. K. Dey, J. Giesen, S. Goswami, and W. Zhao. Shape dimension and approximation from samples. In *Proc. 13th ACM-SIAM Sympos. Discrete Algorithms*, pages 772–780, 2002.

[13] G. Elekes. A geometric inequality and the complexity of computing volume. *Discrete Comput. Geom.*, 1:289–292, 1986.

[14] Z. Furedi and I. Barany. Computing the volume is difficult. In *Proc. 18th Annu. ACM Sympos. Theory Comput.*, pages 442–447, 1986.

[15] J. Giesen and U. Wagner. Shape dimension and intrinsic metric from samples of manifolds with high co-dimension. In *Proc. 19th Annu. ACM Sympos. Comput. Geom.*, pages 329–337, 2003.

[16] J. M. Hammersley. Monte Carlo methods for solving multivariable problems. *Ann. New York Acad. Sci.*, 86:844–874, 1960.

[17] S. Har-Peled and K. R. Varadarajan. Projective clustering in high dimensions using core-sets. In *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, pages 312–318, 2002.

[18] A. Ihler and M. Mandel. Kernel density estimation toolbox for matlab. In *http://www.ics.uci.edu/ ihler/code/*.

[19] S. Lafon. *Diffusion Maps and Geodesic Harmonics*. Ph.D. thesis, Yale University, 2004.

[20] X. Li, W. Wang, R. R. Martin, and A. Bowyer. Using low-discrepancy sequences and the Crofton formula to compute surface areas of geometric models. *Computer-Aided Design*, 35(9):771–782, 2003.

[21] Y.-S. Liu, J.-H. Yong, H. Zhang, D.-M. Yan, and J.-G. Sun. A quasi-Monte Carlo method for computing areas of point-sampled surfaces. *Computer-Aided Design*, 38(1):55–68, 2006.

[22] S. Rosenberg. *The Laplacian on a Riemannian Manifold: An Introduction to Analysis on Manifolds*. Cambridge University Press, 1997.

[23] L. A. Santaló. Integral geometry. In *Chern SS, editor. Studies in global geometry and analysis*, pages 147–95. Washington, DC: The Press of The Mathematical Association of America, 1967.