**Stochastic Heat Kernel Estimation on Sampled Manifolds**

T. Aumentado-Armstrong and K. Siddiqi

1School of Computer Science and Centre for Intelligent Machines, McGill University, Canada

**Abstract**

The heat kernel is a fundamental geometric object associated to every Riemannian manifold, used across applications in computer vision, graphics, and machine learning. In this article, we propose a novel computational approach to estimating the heat kernel of a statistically sampled manifold (e.g. meshes or point clouds), using its representation as the transition density function of Brownian motion on the manifold. Our approach first constructs a set of local approximations to the manifold via moving least squares. We then simulate Brownian motion on the manifold by stochastic numerical integration of the associated Ito diffusion system. By accumulating a number of these trajectories, a kernel density estimation method can then be used to approximate the transition density function of the diffusion process, which is equivalent to the heat kernel. We analyse our algorithm on the 2-sphere, as well as on shapes in 3D. Our approach is readily parallelizable and can handle manifold samples of large size as well as surfaces of high co-dimension, since all the computations are local. We relate our method to the standard approaches in diffusion geometry and discuss directions for future work.


**1. Introduction**

Many sets of data can be considered as samples from Riemannian manifolds embedded in a higher dimensional space. This includes the point clouds and meshes that underpin computer graphics and 3D machine vision, as well as datasets in machine learning considered from a differential geometric viewpoint. For shape-theoretic applications in particular, the Laplace-Beltrami operator (LBO) on a manifold is a very useful tool, as geometric signatures based on the LBO spectrum are invariant to isometric transformations, vary continuously with deformation, have considerable discriminative power, and can be made scale-invariant [RWP06].

Associated to every Riemannian manifold is a mathematical object called the heat kernel, which can be viewed as the fundamental solution to the heat equation on the manifold or as the transition density function of Brownian motion on the manifold. The former representation, as a partial differential equation (PDE), models the ensemble properties of the latter, as a stochastic process. The heat kernel has been used extensively for feature extraction, particularly in a form known as the heat kernel signature [SOG09], in many applications, such as shape recognition [DLL*10, BK10] and correspondence computation [OMMG10]. It also appears in manifold learning, including spectral embedding via diffusion maps [CLL*05] and intrinsic local coordinate construction [JMS08].

Typically, the heat kernel is computed via a spectral decomposition, which requires estimating the eigenvalues and eigenfunctions of the LBO. To this end, substantial literature exists for meshes [BSW08, RWP06, RBG*09] and point clouds [LPG12, BSW09]. However, this form of global computation can be costly, both in memory and time, which has motivated some studies on faster approximations of the heat kernel. Vaxman et al. [VBCG10] devised a multi-resolution approach to estimate the heat kernel, with an approximation to its matrix exponential form. Using the fact that the locality of diffusion at small times implies sparsity of the heat kernel, a low-resolution version of the mesh can be used for estimation, followed by mapping back to the original model. This approach is particularly effective for computing the diagonal of the heat kernel (i.e. the heat kernel signatures). A more recent methodology by Patané and Spagnuolo [PS13] performs spectrum-free estimation of the heat kernel by Chebyshev approximation. Using efficient matrix computations and sparse linear system solvers, the algorithm can function without a mesh structure and avoids the need for multi-resolution prolongation operations.

Herein, we consider the heat kernel from a different viewpoint, using the stochastic calculus and the theory of Ito diffusions. We derive a novel probabilistic algorithm for approximating the heat kernel on a sampled manifold, by estimating the transition densities of diffusion processes on the manifold. The algorithm proceeds by constructing a set of local surfaces and coordinates via moving least squares, over which a diffusion process can be generated by stochastic numerical integration. Kernel density estimation (KDE) over the manifold is then used to approximate the heat kernel.
We begin by providing background on Riemannian geometry and stochastic calculus on manifolds, which underlie our approach. We then develop the details of the algorithm. Results on the 2-sphere are presented and analyzed, followed by examples using points sampled from the surfaces of shape models. We end with a discussion of our method, as well as directions for future work.

2. Theoretical Background

2.1. The Heat Kernel on Riemannian Manifolds

Let $(\mathcal{M}, g)$ be a manifold $\mathcal{M}$ with an associated metric tensor $g$. Further denote the inverse metric $g^{-1}$ to have components $g^{ij}$. Working in local coordinates $x^i$, the metric tensor encodes the curvature of the manifold through the Christoffel symbols of the second kind $\Gamma_{ij}^k$. The Christoffel symbols can be used to define the type-(1,3) Riemann curvature tensor $R^k_{ijk}$, the Ricci curvature tensor $R_{ij} = R^k_{ikj}$, and the Ricci scalar curvature $R = g^{ij}R_{ij}$, which plays a direct role in affecting the diffusion behaviour on the manifold. Note that we use the Einstein summation convention throughout the paper.

Moreover, based on its metric tensor, every Riemannian manifold has an associated LBO, which is an intrinsically defined differential operator denoted $\Delta$. In local coordinates, the LBO can be written:

$$\Delta g f = \frac{1}{\sqrt{\det(g)}} \frac{\partial}{\partial x^i} \left( \sqrt{|\det(g)|} g^{ij} \frac{\partial f}{\partial x^j} \right)$$

The spectrum of the LBO is the set of eigenvalues of the operator $\Delta$; that is, the solutions to the Laplacian eigenvalue problem $-\Delta \phi = \lambda \phi$, where $\lambda \in \mathbb{R}$ satisfies $0 \leq \lambda_k \leq \lambda_l$ for any $\ell > k$ and the $\phi_i : \mathcal{M} \to \mathbb{R}$ are the associated eigenfunctions of the eigensystem, for $i \in \mathbb{Z}_{\geq 0}$ [Ros97, LLWZ12].

Next, consider the heat equation on $(\mathcal{M}, g)$, given by

$$\frac{\partial}{\partial t} u(x,t) = \Delta_g u(x,t)$$

where $x \in \mathcal{M}$ and $t \in \mathbb{R}_{>0}$. The heat kernel $K_t(x,y)$ is the fundamental solution to the heat equation on the manifold, but can also be interpreted as the amount of some substance (e.g. heat) that is transferred from $x$ to $y$ in time $t$ via diffusion. Computationally, the heat kernel is generally calculated via its spectral representation:

$$K_t(x,y) = \sum_{i=0}^{\infty} \exp(-\lambda_i t) \phi_i(x) \phi_i(y)$$

(1)

using the eigenpairs of the LBO. The autodiffusion function $K_t(x) = K_t(x,x)$ can then be used for feature extraction, constructing a vector signature at each $x \in \mathcal{M}$ called the heat kernel signature (HKS) [SO90, GBAL09]. Intuitively, the HKS describes how heat flows out from a point over time, capturing the multiscale effect of the manifold’s geometry on the diffusion process. For small $t$, the autodiffusion can be expanded as [U87, MS67]

$$K_t(x) = \frac{1}{(4\pi t)^d/2} \left( 1 + \frac{R(x)}{6} t + O(t^2) \right)$$

(2)

for a manifold of dimension $d$ in a $D$-dimensional ambient space, exemplifying the role of the Ricci curvature in affecting local diffusion movement. The importance of this term suggests higher order surface approximations, allowing estimation of the local curvature, are important for accuracy, at least on short spacetime scales.

2.2. Brownian Motion on Riemannian Manifolds

In stochastic terms, the heat kernel is directly related to the probability density of a particle following Brownian motion on the manifold. This can be seen as follows [It62, IW14]: let $x \in \mathcal{M}$ be the starting point of a Brownian motion $X_t$ on $\mathcal{M}$ and $f : \mathcal{M} \to \mathbb{R}$ be a scalar field. Then the expectation $u(t, x) = \mathbb{E}_x[f(X_t)]$ satisfies a heat equation on the manifold, given by $\frac{\partial}{\partial t} u = \frac{1}{4} \Delta_g u$, $u(0, x) = f(x)$. One can interpret this to mean that the “average”, or expected, behaviour of an ensemble of Brownian motions on the manifold evolves according to the heat equation. More specifically, in local coordinates, Brownian motion on $(\mathcal{M}, g)$ can be written as a system of stochastic differential equations (SDEs) in the Itô sense, via [It62, Hsu02]

$$dX^i_t = \sigma^i_j(X_t) dB^j_t - \frac{1}{2} \sigma^i_j(X_t) \sigma^j_i(X_t) dt$$

(3)

where $X_t$ describes a $d$-dimensional Itô diffusion process on $\mathcal{M}$, $\sigma^i_j$ is a $d$-dimensional Wiener process, $\mu = g^{ij} \Gamma^k_{ij}$, and $\sigma = \sqrt{g^{-1}}$. The infinitesimal generator of $X_t$, which encodes the behaviour of the diffusion process over infinitesimal time intervals, is given by $\mathcal{L} = \Delta_g / 2$. Generally, the matrix field $\sigma$ is called the diffusion coefficient, while the vector field $\mu$ is called the drift coefficient. Note that if $g$ is symmetric positive definite (SPD), so too is $g^{-1}$, and that SPD matrices have a unique SPD matrix root, making $\sigma$ well-defined.

The transition density function of the Brownian motion $X_t$ can be written as $p(x,t|y)$, since an Itô diffusion has the Markov property, which describes the probability distribution of the stochastic process reaching $x$ after time $t$ when starting from $y$. It can be shown [Hsu02] that the fundamental solution (i.e. minimal heat kernel) of the equation $(\partial_t - \mathcal{L}) u = 0$ is the transition density of $X_t$. Note that the factor of $1/2$ corresponds to a scaling of time; here, we modify $\mu$ and $\sigma$ such that $K_t(x,y) = p(x,t|y)$, so that computations from eqs. 1 and 2 match those from our stochastic approach.

A natural approach to estimating the transition density is to use the Fokker-Planck (or Kolmogorov Forward) equation, which describes the evolution of the probability density over time:

$$\frac{\partial p(x,t|y)}{\partial t} = -\frac{\partial}{\partial x}[\mu^i(x)p(x,t|y)] + \frac{1}{2} \frac{\partial^2}{\partial x^i \partial x^j} [\sigma^i_j(x)p(x,t|y)]$$

where $y$ is the fixed start point of the process. The problem would then be reduced to solving this PDE on the sampled manifold. Here, however, we choose to investigate the properties of a stochastic formulation and estimation methodology instead. One reason for this is the difficulty encountered in solving PDEs on high dimensional point sets, particularly in the case of high co-dimension, though recent works have focused on this issue [LZ13]. We are motivated in part by the success of probabilistic Monte Carlo methods over deterministic approaches to numerical integration in certain cases.
2.3. Transition Density Estimation of Ito Diffusions

We next consider the problem of estimating the transition density of a continuous space-time stochastic diffusion process from an approximation of the process (i.e. a trajectory generated by numerical integration). Let \( h \) be the step size of numerical integration, for a simulated process \( X_t \) starting from \( x \), which approximates \( X \). It can be shown that the transition density of \( X_t \) converges to the true transition density as \( h \) tends to zero [BT96, MSS'04]. Furthermore, in \( \mathbb{R}^d \), previous studies [KH'97, HW'96] showed that

\[
\hat{p}_h(x,t|y) = \mathbb{E} \left[ \delta_h (\tilde{X}_t - y) \right]
\]

where \( \delta_h(a) = (2\pi h^2)^{-d/2} \exp \left( -|a|^2/(2h^2) \right) \), converges to the true transition density as \( h \) shrinks as well. This suggests a natural estimator using the means of a number of generated trajectories, but one can more generally use KDE to approximate the transition density [MSS'04].

However, the fact that our KDE takes place on a Riemannian manifold induces extra challenges. In particular, since we assume access only to a set of point samples of \( M \), density estimators that rely on computing certain differential geometric properties of the manifold (e.g. geodesic distances, volume densities) may be difficult to use. Thus, we instead adopt the approach of Ozakin and Gray [OG09] for KDE on Riemannian submanifolds of Euclidean space. Let \( d \) be the dimension of the submanifold \( M \) and \( D \) be the that of the ambient Euclidean space. The idea is to use the fact that sufficiently close points on \( M \) should have intrinsic distances similar to those of \( \mathbb{R}^D \), but that the kernel should be normalized for \( \mathbb{R}^d \), since at small enough scales, \( M \) should look like \( \mathbb{R}^d \). Therefore, the KDE can be approximated by:

\[
\tilde{K}_t(x,y) = \frac{1}{n_T} \sum_{j=1}^{n_T} \Psi \left( ||\hat{X}^{(i)}_{t,j} - y||_E \right) (4)
\]

where \(| \cdot |_E\) denotes Euclidean distance in the ambient space, \( n_T \) is the number of generated trajectories, \( \hat{X}^{(i)}_{t,j} \) is \( i \)-th trajectory starting from \( x \) at time \( t \), \( \Psi \) is a kernel function, and \( \delta_h \) is the kernel bandwidth.

Finally, we note that a useful first-order local approximation of the transition density of the SDE (3) is given by

\[
\hat{p}(x,t|y) = \phi(x) \cdot \mu(y) |t| \sigma^2(y) |y| (5)
\]

where \( \phi \) is the multivariate Gaussian density with mean and covariance parameters dependent on the drift \( \mu \) and diffusion \( \sigma \) coefficient fields of the Ito process [DG02]. Notice that this approximation (eq. 5) recovers the discretization of Belkin and Niyogi [BN03], if one assumes normal coordinates about \( y \). In this case, if time is scaled to solve \( 2\mathcal{L}u = \frac{\partial u}{\partial t} \), we get \( \hat{p}_h(x,t|y) = (4\pi h)^{-d/2} \exp \left( -||x-y||_E^2 \right) \), which is valid at sufficiently small spacetime scales.

3. A Probabilistic Algorithm for Heat Kernel Estimation

3.1. Outline of Algorithm

Our algorithm consists of three main steps: (1) constructing a set of local surfaces approximating the manifold, (2) simulating Brownian motion in local coordinates across these surfaces, and (3) estimating the heat kernel from these stochastic trajectories.

The first step is solved by the moving least squares (MLS) approach [LS81], which allows constructing a local surface around each point based on distance-weighted least squares regression fitting of a multivariate polynomial. This permits the definition of a local coordinate system and parametric surface around each point on the sampled manifold.

For the second step, we simulate Brownian motion via stochastic numerical integration of eq. 3 in local coordinates. The challenge is to move between local surface parametrizations (i.e. transitioning between charts as the process moves across the manifold). We propose to solve this by “jumping” from local surface to surface, by iteratively alternating between simulating the SDE for a time \( T_h \) and projecting the process onto a close local surface. Note that the trajectory always lies on a local surface approximation; any discrepancies caused by the “jump” are due to differences between the starting and ending MLS surfaces. Intuitively, if the local surfaces are smoothly changing over a sufficiently well-sampled manifold, a trajectory \( X_t \) should lie close to the true surface as \( T_h \) shrinks. Consequently, this method allows simulation without attempting to construct a global parametrization of the data, which is itself a difficult problem in computational manifold learning (see e.g. [Bra03]).

Finally, we use KDE, with bandwidth chosen by calibration with respect to local analytic expansion of the transition density, to estimate the heat kernel.

We present pseudocode in algorithm 1, where \( P \) is the set of samples from the manifold and \( S \subset P \) is a subset of sample points of interest for which we wish to compute the heat kernel. While the algorithm can be generalized to higher dimensions, we describe it here for the specific case of a 2D surface in 3D space.

Algorithm 1 Stochastic Heat Kernel Estimation Algorithm

1: procedure HEATKERNELESTIMATE\((P,S)\)
2: for \( p \in P \) do
3: Construct MLS surface \( \Lambda_p \)
4: end for
5: for \( s \in S \) do
6: for \( i \in [0,n_T] \) do
7: \( q \leftarrow p \)
8: while \( t < T_{\text{max}} \) do
9: \( T_h \) steps do
10: Integrate \( X^{(i)}_{p,t+h} \) to \( X^{(i)}_{p,t+h} \) on \( \Lambda_q \)
11: end for
12: \( q \leftarrow \arg\min_{q\in P} ||X^{(i)}_{p,t+h} - q|| \)
13: end while
14: end for
15: end for
16: for \( p \in S \) do
17: for \( q \in P \) do
18: \( K_t(p,q) = \frac{1}{n_T} \sum_{k=1}^{n_T} \Psi \left( \frac{||X^{(k)}_{p,t+h} - q||_E}{\delta_h} \right) \)
19: end for
20: end for
21: return \( K \)
22: end procedure
3.2. Local Manifold Approximation via Weighted MLS

Inspired by the approach by Liang et al. [LZ13,LLWZ12], we construct a set of local coordinates and MLS surfaces around each point. Note that the importance of local curvature (see sec. 2.1 and 2.2) suggests that linear approximations to a manifold will significantly reduce accuracy (particularly on short spacetime scales), since no curvature term will be present. As such, we use a second-order bivariate polynomial surface fitted locally around each point, via weighted least squares.

First, we construct a local coordinate system for each point \( p \in P \). Let \( N_k(p) \) denote the \( k \) closest points to \( p \), by some distance metric (e.g. graph distance on meshes, Euclidean distance on point clouds). Note that \( k \) may depend on \( p \) (e.g. using the \( s \)-ring on a mesh). Via principal component analysis of \( \mathcal{N}_k(p) \), we obtain the normalized eigenvectors of the covariance matrix: the first two, \( \theta_{p,1} \) and \( \theta_{p,2} \), span the local tangent space of the point, while the last \( \theta_{p,3} \) approximates the local surface normal. We then take \( p \) as the origin and \( (\theta_{p,1}, \theta_{p,2}, \theta_{p,3}) \) as the axes of the local coordinate system at \( p \).

Next, we construct a surface \( \mathcal{S}_p(x,y) \) in local coordinates, by fitting to \( \mathcal{N}_k(p) \) via weighted least squares. To ensure the presence of \( \mathcal{S}_p \), we use a second-order approximation:

\[
\mathcal{S}_p(x,y) = y_0 + y_1 x + y_2 y + y_3 x^2 + y_4 x y + y_5 y^2
\]

where we estimate the parameters \( y_i \) by weighted least squares minimizing the loss function:

\[
L(p) = \sum_{q \in N_k(p)} W(||p - q||_2) [\mathcal{S}_p(q_x, q_y) - q_z]^2
\]

where \( q = (x_q, y_q, z_q) \) in local coordinates, and we chose \( W(d) = \frac{1}{(d^2 + \epsilon)} \) with the 2-norm distance metric. The local surface about \( p \) can thus be represented by \( \mathcal{S}_p = (x,y,\mathcal{S}_p(x,y)) \) in local coordinates, giving a degree 2 bivariate polynomial surface on which Christoffel symbols may be computed. Note that the Ricci scalar is then given by \( \mathcal{S}_p \) across the local surface. The main benefit of this approach is that it reduces the high computational cost of evaluating \( \mu \) and \( \sigma \) during trajectory generation. Note that we are not approximating \( \mathcal{S}_p \) in any way, and the process always stays on the full polynomial surface, including its non-linear terms; instead, only the local, small-scale behaviour of the diffusion process is changed. In this work, we fix \( \tau \) to be 5% of \( \tau_{\text{max}} \), unless otherwise noted.

The question remains as to how to move between local charts and surfaces during the diffusion process. Starting from a point \( p \), we simulate the Ito diffusion for \( \tau = T_X \) steps, with step-size \( \delta t \), ending up at a point \( X_t \). Let \( q \in P \) be the closest point to \( X_t \), and denote \( \mathcal{S}_q = (x_q, y_q, z_q) \) as the position vector of \( X_t \) in the local coordinates of \( q \). By construction, \( X_t \) is on \( \mathcal{S}_q \), but this may not correspond to a position on \( \mathcal{S}_p \). Therefore, we project \( X_t \) onto \( \mathcal{S}_p \) before continuing the numerical integration process. This is done by computing:

\[
\tilde{x}, \tilde{y} = \arg \min_{x,y} \sqrt{\sum (x-x_q)^2 + (y-y_q)^2 + (z(z(x,y) - z_q))^2}
\]

using the Broyden-Fletcher-Goldfarb-Shanno numerical optimization algorithm to compute \( \tilde{x}, \tilde{y} \) [Avr03]. As such, \( \tilde{X}_t = (\tilde{x}, \tilde{y}, z_{\mathcal{S}_p}(\tilde{x}, \tilde{y})) \) is the closest point to \( X_t \) on \( \mathcal{S}_p \). We then restart the diffusion process from \( \tilde{X}_t \).

Note that \( T_X \) is essentially the time between “jumps”; i.e. between moves to a new \( \Lambda_p \). Importantly, a given trajectory is always guaranteed to be on some MLS surface \( \Lambda_p \); however, as the MLS approximation is a less reliable representation of \( M \) far away from \( p \), if \( T_X \) is large, the projection distance \( D_q(X_t) = ||X_t - \tilde{X}_t|| \) may also be large. Further, since the nearest neighbour determines the chart choice, it is possible that a small perturbation in \( X_t \) can change the choice of \( \Lambda_q \), thus altering \( D_q(X_t) \) as well. In this sense, like the integration step-size \( \delta t \), the jump-time parameter \( T_X \) provides a trade-off between computation time and accuracy. For a well-sampled manifold, with small \( T_X \), \( \delta t \) and \( \delta q \) should be close, and \( \Lambda_p \) and \( \Lambda_q \) sufficiently similar that \( D_q(X_t) \) should be small, thus helping to preserve the continuity of the approximate diffusion process.

3.4. Transition Density Estimation with KDE

Using the procedure above, we may repeatedly generate stochastic trajectories on the manifold, which can then be used to estimate the transition density via KDE using eq. 4. Note that we can also use the symmetry of the heat kernel to obtain a better estimate by averaging, if both \( p, q \in S \). Further, for densely sampled manifolds, we can estimate the heat kernel for only a proportion \( r \) of the points and interpolate to the rest; we let \( r = 1 \) unless otherwise specified.

Here, we use a Gaussian kernel, written \( \psi(z) = \exp(-||z||^2)/\sqrt{2\pi} \). However, we must still determine the kernel bandwidth \( \delta_r \), which is known to be a difficult problem [MSS04]. Based on theoretical considerations, we define the kernel bandwidth as \( \delta_r = \eta_m \delta_r \), which starts the bandwidth at
approximately the optimal order of magnitude [OG09, MSS’04]. Hence, our parameter search is over \( \eta_h \), to choose the kernel bandwidth \( \delta_h \). Since we do not have the ground-truth values for the heat kernel, we use the short-term analytic expansion of the transition density as a multivariate Gaussian (eq. 5) to calibrate \( \eta_h \). Alternatively, one could use eq. 2.

To do this, we define the mean squared logarithmic error (RMSE), defined by
\[
E(a, b) = \frac{1}{2} \sum R \ln(a_i + 1) - \ln(b_i + 1)]^2,
\]
where \( a, b \in \mathbb{R}^n \), which is useful for comparing sequences with high variations in magnitude. We then use the estimated autodiffusion function \( \tilde{K}_t(x) \) and the first-order approximation of eq. 5, \( K_t(x) \), at several small discretely sampled times, to search for the \( \eta_h \) that minimizes \( \tilde{E}(\tilde{K}_t(x), K_t(x)) \).

3.5. Algorithmic Complexity
Consider a point set \( P \) with \( N = |P| \) and a set of points of interest \( S \subset P \). We consider the time complexity of algorithm 1, focusing on the low dimensional case (both intrinsic \( d \) and ambient \( D \) small). The first step of the algorithm is local MLS surface construction. Let \( \tilde{S} \) be the cost of finding \( \tilde{N}_p(p) \) for \( p \in P \). In general, \( \tilde{S} \) can depend on \( N \) (e.g. \( \sim O(\log(N)) \) with a KD-tree), but it does not have to (e.g. using the rings of a node in a mesh). For large \( N \) and \( D \), using approximate nearest neighbour approaches would be preferable. Estimation of local coordinate frames and least squares fitting depend only on \( k, d, \) and \( D \), all of which we assume to be small. Hence, the overall time cost is \( O(N \tilde{S}) \), and the memory cost is \( O(N) \), since only a constant number of values per point is stored.

The second step is trajectory generation. Let \( C_\ell \) be the cost of a single stochastic integration step with full curvature computations (i.e. \( t \leq \tau \)), and \( C_L \) that with approximated curvature (i.e. \( t > \tau \)). Both constants depend only on the algorithm used and \( d \), but \( C_\ell > C_L \). Further, let \( S \) be the cost to find the closest neighbour \( p \in \tilde{P} \) to a point \( q \in \tilde{P} \) (e.g. \( \sim \log(N) \) using a KD-tree). Then, the overall cost to generate \( n_T \) sample trajectories, for a single \( p \in \tilde{S} \), is \( O(\tilde{T}n_T \log(T_{\text{max}}/T_h + \tau(C_\ell + [T_{\text{max}} - \tau(C_L)/h])) \). This is sufficient to estimate a single row and column in the full heat kernel. Note that the two terms correspond to the cost of jumping and integrating, respectively, and that the dependence on \( N \) can be sublinear, while the dependence on most other parameters is linear. Further, let \( H_T \) be a set of times of interest (i.e. we consider only \( K_t \) with \( t \in H_T \)). If only the simulated points needed for \( H_T \) are stored, the total memory cost across \( S \) is \( O(\tilde{T}|n_T| |H_T|) \).

The third step is KDE on the manifold. Ignoring the linear dependence on \( D \) and supposing we consider only the submatrix of the heat kernel given by \( S \), so that \( K_t \in \mathbb{R}^{\tilde{S} \times N} \), then the total cost of computing the heat kernel is \( O(n_T |S| |N| |H_T|) \). Essentially, this is the cost of computing \( |S| \) rows and columns of the heat kernel across all \( N \) other points, sampled at \( |H_T| \) different times. In terms of memory storage, this costs \( O(|S| |N| |H_T|) \).

4. Empirical Results
All methods were implemented in Python, using NumPy and SciPy [OH07], sdeint [Mat17], and the KD-tree in Scikit-learn [PVG*11].
show examples of the autodiffusion function (left panel) and heat kernel values between antipodal points on the sphere (right panel).

It is also of importance to ensure that the trajectories indeed stay on the manifold. In Figure 3, we plot $|X_t|$ over time, where adherence to the sphere would imply $|X_t| = 1$. The plot demonstrates the importance of $T_8$, as the deviation from the correct distance continuously increases between every jump, but each projection snaps the value back to the manifold. Yet, overall, the deviation is generally less than 0.005 on the sphere with unit radius, suggesting reasonable adherence to the manifold structure.

4.2. Heat Kernel on Shape Models

We next demonstrate our algorithm on mesh models from the TOSCA dataset [BBK07]. To compute a heat kernel against which to compare our method, we use the linear finite element approach [RWP06, RBG09], computing the heat kernel via its spectral representation (eq. 1).

To run our algorithm on the TOSCA models, the only alteration we make from the sphere case is to compute the nearest neighbours during local surface construction as the two-ring around each node. We use $h_0 = 0.01, T_8 = 0.1, n_T = 500, \tau = \infty$, and $T_{max} = 10^5$ with SRK. The resulting heat kernels are fairly similar (see Figure 4); however, in comparison with the spectral method, the stochastic approach tends to have many more infinite times of arrival (the darkest blue) and is more localized, whereas the spectral approach looks more smoothed out. We also considered the fidelity of the process to staying on the manifold (Figure 5). Due to the narrowness and sudden change in curvature of the shark’s fins, it is possible for the SDE to move off the manifold. Such problems do not tend to occur on the more robustly sampled parts of the manifold with lower rates of curvature change.

We also tested our approach on several models with more vertices: buddha and Chinese lion from the AIM@SHAPE repository, and armadillo from the Stanford 3D scanning repository. We do not use face or edge information; i.e. all nearest neighbours are computed with a KD-tree. The parameters were fixed as $k = 15, T_8 = 0.25, h_0 = 0.05, n_T = 200, T_{max} = 500.0$, and $\eta_0 = 14.9$, using SEM. The discrete time set was log-sampled from 0.5 to 500 with $|H_T| = 5$. Given the higher sample density, we also let $\ell = 3$ and $r = 0.25$. Looking at the computational times for these models (Table 1), we see that the cost of trajectory generation changes little with $N$, while that of KDE and MLS increase linearly, as expected. Note that the timing reflects the cost of filling [S] rows of $K$ (each of length $N = |P|$), at $|H_T|$ times. Further, one can see in Figure 6 that the algorithm still produces trajectories close to the manifold, despite the additional approximations.
trolled by altering its parameters, particularly the trade-off between accuracy and computation time to be con-
dently. The cost and stability of our algorithm does not depend on easily parallelizable, since the trajectories are generated indepen-
dence of not requiring global computations (e.g. solving a global eigenvalue problem) and of working locally, mostly in the lower di-
manifold in pathological situations. However, we offer the advan-
tories from this process.

for estimating the heat kernel on the manifold using sample trajec-
a sampled manifold via numerical integration of an SDE, period-
moment of the submanifold. This suggests that our approach may be useful for manifolds of high co-dimension or in cases with an dimension of the submanifold. This suggests that our approach may

Table 1: Columns: model size $N$, number of feature points $|S|$, and computation time for MLS surface construction (including KD-
aka of the heat kernel, while we have computed selected rows and columns. In its present form, our implementation is slower than [PS13, VBCG10], but considerable opportunities exist for im-

5. Discussion

Our focus has been on the probabilistic interpretation of the heat kernel and its connection to the Ito calculus on manifolds. We first devised an algorithm for approximating Brownian motion on a sampled manifold via numerical integration of an SDE, periodic-

The main limitations of the algorithm in its current form are its computational cost and the potential for trajectories to stray off the manifold in pathological situations. However, we offer the advantage of not requiring global computations (e.g. solving a global eigenvector problem) and of working locally, mostly in the lower di-

Another direction for future investigation is the adaptation of our approach to handle other SDEs on manifolds. This may be helpful for the extraction of other shape theoretic signatures, in the spirit of the famous Feynman-Kac formula relating certain classes of PDEs to stochastic Ito processes. While we do not prove convergence of our algorithm, most of the substeps have known convergence rates or approximation error bounds that improve with data or computa-

Ultimately, we hope that this alternative stochastic viewpoint on diffusion geometry may prove useful in the future, as an inspiration for new algorithms in manifold-theoretic approaches to geometry processing.

Acknowledgements This work was supported by funding from the Natural Sciences and Engineering Research Council of Canada.
References


© 2017 The Author(s)
Computer Graphics Forum © 2017 The Eurographics Association and John Wiley & Sons Ltd.