DOI: 10.1111/cgf.13251

Eurographics Symposium on Geometry Processing 2017
J. A. Bærentzen and K. Hildebrandt
(Guest Editors)

Volume 36 (2017), Number 5

Stochastic Heat Kernel Estimation on Sampled Manifolds

T. Aumentado-Armstrong[1] and K. Siddiqi[1]

[1]School 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.

Categories and Subject Descriptors (according to ACM CCS): G.3 [Mathematics of Computing]: Probability and Statistics—Probabilistic Algorithms, I.3.5 [Computer Graphics]: Computational Geometry and Object Modelling—Geometric Algorithms, Languages, and Systems

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 analysed, 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 $\Gamma^i_{k\ell}$. The Christoffel symbols can be used to define the type-(1,3) Riemann curvature tensor R^ℓ_{ijk}, the Ricci curvature tensor $R_{ij} = R^{\mathrm{p}}_{i\mathrm{p}j}$, and the Ricci scalar curvature $\mathcal{R} = g^{ij}R_{ij}$, 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 Δ_g. 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 Δ_g; that is, the solutions to the Laplacian eigenvalue problem $-\Delta_g \phi_i = \lambda_i \phi_i$ where $\lambda_i \in \mathbb{R}$ satisfies $0 \leq \lambda_k \leq \lambda_\ell$ 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:

$$K_t(x,y) = \sum_{i=0}^{\infty} \exp(-\lambda_i t) \phi_i(x) \phi_i(y) \tag{1}$$

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) [SOG09, 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 [U*87, MS67]

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

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 [Itô62, 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 $\frac{\partial}{\partial t} u = \frac{1}{2}\Delta_g u$, $u(0,x) = f(x)$. 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 Ito sense, via [Itô62, Hsu02]

$$dX^i_t = \sigma^i_j(X_t)\, dB^j_t - \frac{1}{2}\mu^i(X_t)\, dt \tag{3}$$

where X_t describes a d-dimensional Ito diffusion process on $\mathcal{M}$, B_t is a d-dimensional Wiener process, $\mu^i = g^{jk}\Gamma^i_{jk}$, 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 σ is called the diffusion coefficient, while the vector field μ 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 σ well-defined.

The transition density function of the Brownian motion X_t can be written as $p(x,t|y)$, since an Ito 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 μ and σ 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:

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

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_s be the step size of numerical integration, for a simulated process $\tilde{X}_t$ starting from x, which approximates X_t. It can be shown that the transition density of $\tilde{X}_t$ converges to the true transition density as h_s tends to zero [BT96, MSS*04]. Furthermore, in $\mathbb{R}^d$, previous studies [KH*97, HW*96] showed that

$$\tilde{p}_h(x,t|y) = \mathbb{E}\left[\tilde{\phi}_{h_s}(\tilde{X}_t - y)\right]$$

where $\tilde{\phi}_h(a) = (2\pi h^2)^{-d/2}\exp(-|a|^2/[2h^2])$, converges to the true transition density as h_s 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 $\mathcal{M}$, 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 $\mathcal{M}$ and D be the that of the ambient Euclidean space. The idea is to use the fact that sufficiently close points on $\mathcal{M}$ 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, $\mathcal{M}$ should look like $\mathbb{R}^d$. Therefore, the KDE can be approximated by:

$$\widehat{K}_t(x,y) = \frac{1}{n_T \delta_h^d} \sum_{j=1}^{n_T} \Psi\left(\frac{||\tilde{X}_{x,t}^{(j)} - y||_E}{\delta_h}\right) \tag{4}$$

where $||\cdot||_E$ denotes Euclidean distance in the ambient space, n_T is the number of generated trajectories, $\tilde{X}_{x,t}^{(i)}$ is ith trajectory starting from x at time t, Ψ is a kernel function, and δ_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

$$\widehat{p}(x,t|y) = \phi(x\,|\,y + \mu(y)t, \sigma^2(y)t) \tag{5}$$

where ϕ is the multivariate Gaussian density with mean and covariance parameters dependent on the drift μ and diffusion σ 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 $2\mathcal{L}u = \frac{\partial}{\partial t}u$, we get $\widehat{p}_n(x,t|y) = (4\pi t)^{-d/2}\exp\left(\frac{-||x-y||_2^2}{4t}\right)$, 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_δ 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_δ 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

procedure HEATKERNELESTIMATE(P,S)
  for $p \in P$ **do**
    Construct MLS surface Λ_p
  end for
  for $p \in S$ **do**
    for $i \in [0, n_T]$ **do**
      $q \leftarrow p$
      while $t < T_{\max}$ **do**
        for T_δ/h_s steps **do**
          Integrate $X_{p,t}^{(i)}$ to $X_{p,t+h_s}^{(i)}$ on Λ_q
        end for
        $q \leftarrow \underset{q \in P}{\operatorname{argmin}} ||X_{p,t}^{(i)} - q||$
      end while
    end for
  end for
  for $p \in S$ **do**
    for $q \in P$ **do**
      $K_t(p,q) = \frac{1}{n_T \delta_h^d} \sum_{k=1}^{n_T} \Psi\left(\frac{||X_{p,t}^{(k)} - q||_E}{\delta_h}\right)$
    end for
  end for
  return K
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 $\mathcal{N}_k(p)$ 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, $e_{p,1}$ and $e_{p,2}$, span the local tangent space of the point, while the last $e_{p,3}$ approximates the local surface normal. We then take p as the origin and $(e_{p,1}, e_{p,2}, e_{p,3})$ as the axes of the local coordinate system at p.

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

$$z_p(x,y) = \gamma_0 + \gamma_1 x + \gamma_2 y + \gamma_3 x^2 + \gamma_4 xy + \gamma_5 y^2$$

where we estimate the parameters γ_i by weighted least squares minimization of the loss function:

$$L(p) = \sum_{q \in \mathcal{N}_k(p)} W(||p-q||_2)[z_p(x_q,y_q) - z_q]^2$$

where $q = (x_q, y_q, z_q)$ in local coordinates, and we chose $W(d) = (d^2+\varepsilon)^{-1}$ with the 2-norm distance metric. The local surface about p can thus be represented by $\Lambda_p = (x, y, z_p(x,y))$ in local coordinates, giving a degree 2 bivariate polynomial surface on which an Ito diffusion process may be simulated. When the manifold is densely sampled, one can increase computational efficiency by interpolating Λ_p to its $k/\ell - 1$ closest neighbours; we let $\ell = k$ unless otherwise specified.

3.3. Stochastic Integration and Chart Transition

For 2D surfaces in 3D, the representation of Λ_p as a local parametric surface allows us to use the classical differential geometry of surfaces. The tangent vectors of Λ_p are then $v_{p,1} = \partial_x \Lambda_p$ and $v_{p,2} = \partial_y \Lambda_p$. The local metric tensor is then given by the first fundamental form of the surface; i.e. $g_{ij} = v_{p,i} \cdot v_{p,j}$, from which the Christoffel symbols may be computed. Note that the Ricci scalar curvature is simply twice the Gaussian curvature of Λ_p, so that $\mathcal{R} = 2[\partial_{xx}\partial_{yy} z_p - (\partial_{xy} z_p)^2]/[1 + (\partial_x z_p)^2 + (\partial_y z_p)^2]^2$. Hence, at a given point X_t on the 2D surface, we may compute $\mu(X_t)$ and $\sigma(X_t)$, for use in a stochastic numerical integration scheme applied to eq. 3. We considered two integration methods for approximating solutions of SDEs: a stochastic Runge-Kutta algorithm (SRK) [Röß10] and the Euler-Maruyama algorithm (SEM) [Mar55].

However, although the curvature terms of the manifold are determining factors of the heat kernel at short spatiotemporal scales (e.g. see eq. 2), this importance is overshadowed by the variance of the processes at larger scales (i.e. if one is interested in the point p, the small-scale behaviour of diffusion far away from p is less important). This is akin to the observation by Vaxman et al. that fine-grained geometric features are less important at higher time scales [VBCG10]. Thus, given a threshold τ, at high $t > \tau$ times, we simulate the process on the surface Λ_p as if it were uncurved. This is equivalent to using the metric tensor of the linear approximation to Λ_p at p across the local surface. The main benefit of this approach is that it reduces the high computational cost of evaluating μ and σ during trajectory generation. Note that we are not approximating Λ_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 τ to be 5% of $t_{\max}$, 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 $n_s = T_\delta / h_s$ steps, with step-size h_s, ending up at a point X_t. Let $q \in P$ be the closest point to X_t, and denote $\tilde{X}_t = (\tilde{x}, \tilde{y}, \tilde{z})$ as the position vector of X_t in the local coordinates of q. By construction, X_t is on Λ_p, but this may not correspond to a position on Λ_q. Therefore, we project X_t onto Λ_q before continuing the numerical integration process. This is done by computing:

$$\hat{x}, \hat{y} = \underset{x,y}{\operatorname{argmin}} \; (x-\tilde{x})^2 + (y-\tilde{y})^2 + (z_q(x,y) - \tilde{z})^2$$

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

Note that T_δ is essentially the time between "jumps"; i.e. between moves to a new Λ_i. Importantly, a given trajectory is always guaranteed to be on some MLS surface Λ_p; however, as the MLS approximation is a less reliable representation of $\mathcal{M}$ far away from p, if T_δ is large, the projection distance $D_q(X_t) = ||X_t - \widehat{X}_t||$ 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 Λ_q, thus altering $D_q(X_t)$ as well. In this sense, like the integration step-size h_s, the jump-time parameter T_δ provides a trade-off between computation time and accuracy. For a well-sampled manifold, with small T_δ, p and q should be close, and Λ_p and Λ_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 $\Psi(z) = \exp(-|z|^2)/\sqrt{\pi^d}$. However, we must still determine the kernel bandwidth δ_h, which is known to be a difficult problem [MSS*04]. Based on theoretical considerations, we define the kernel bandwidth as $\delta_h = \eta_h n_T^{-1/(4+d)}$, which starts the bandwidth at

approximately the optimal order of magnitude [OG09, MSS*04]. Hence, our parameter search is over η_h, to choose the kernel bandwidth δ_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 η_h. Alternatively, one could use eq. 2.

To do this, we define the mean squared logarithmic error (RMSLE), defined by $\mathcal{E}(a,b)^2 = \frac{1}{n}\sum_i [\ln(a_i+1) - \ln(b_i+1)]^2$, 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 $\widehat{K}_t(x)$ and the first-order approximation of eq. 5, $K_t(x)$, at several small discretely sampled times, to search for the η_h that minimizes $\mathcal{E}(\widehat{K}_t(x), K_t(x))$.

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 $\tilde{S}_N$ be the cost of finding $\mathcal{N}_k(p)$ for $p \in P$. In general, $\tilde{S}_N$ can depend on N (e.g. $\sim O(k\log(N))$ 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 $O(N\tilde{S}_N)$, 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_S be the cost of a single stochastic integration step with full curvature computations (i.e. $t \leq \tau$), and C_L be that with approximated curvature (i.e. $t > \tau$). Both constants depend only on the algorithm used and d, but $C_S > C_L$. Further, let S_N be the cost to find the closest neighbour in P to a point $q \notin P$ (e.g. $\sim\log(N)$ using a KD-tree). Then, the overall cost to generate n_T sample trajectories, for a single $p \in S$, is $O(n_T[S_N T_{\max}/T_\delta + (\tau C_S + [T_{\max} - \tau]C_L)/h_s])$. 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 $O(|S|n_T|H_T|)$.

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 $K_t \in \mathbb{R}^{|S|\times N}$, 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 [Oli07], sdeint [Mat17], and the KD-tree in Scikit-learn [PVG*11].

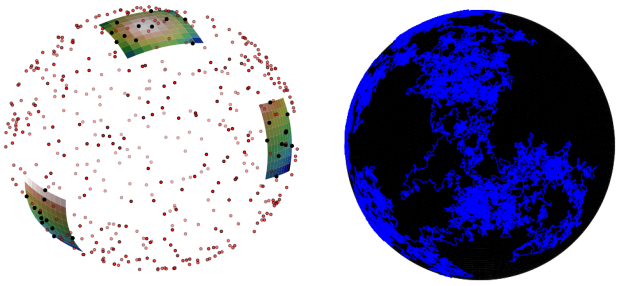

Figure 1: *Visualization of estimation on $\mathbb{S}^2$, with $|P| = 500$. Left: three example local surfaces on the sphere (black denotes points involved in computing the MLS surface). Right: a single example trajectory of Brownian motion on the sphere generated by our algorithm using $h_s = 1.25 \times 10^{-4}$, $\tau = \infty$ with SRK.*

4.1. Heat Kernel on the Sphere

The heat kernel on the sphere $\mathbb{S}^2$ can be computed with its spectral representation (eq. 1). In this case, the eigenfunctions of the unit sphere are the spherical harmonics, and it can be shown [Chu06] that one can compute the heat kernel as:

$$K_t(p,q) = \sum_{i=0}^{\infty} \frac{2i+1}{4\pi} \exp(-i(i+1)t) P_i^0(p \cdot q)$$

where the eigenvalues of Δ_g are $\lambda_i = i(i+1)$ and P_i^j are the associated Legendre functions.

We generate a point cloud sphere by uniformly sampling m points from $\mathbb{S}^2$, using $p_i = v/||v||_2$, where $v_j \sim \mathcal{N}(0,1)$. See Figure 1 for a visualization of the algorithm on the sphere. We then compute our estimated heat kernel $\tilde{K}_t$ with algorithm 1. We fix $T_\delta = 0.05$, $k = 10$, $T_{\max} = 5.0$, and $n_T = 1250$. Heat kernel values are logarithmically sampled from e^{-4} to $T_{\max}$ at 2000 points, and smoothed by a degree-1 Savitzky-Golay filter with window size 1/10 of the signal length (in log-space). Parameter calibration with the first 10% of time values yielded $\eta_h = 0.1887$. In Figure 2, we

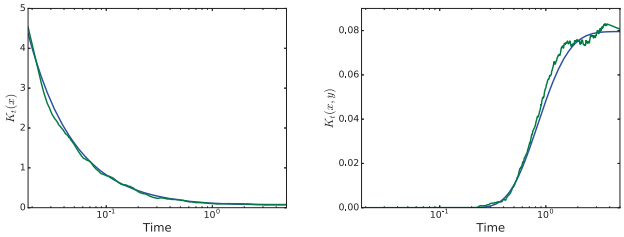

Figure 2: *Heat kernel values over time. Blue: ground truth computed by spherical harmonics; Green: stochastic estimation ($h_s = 5.0 \times 10^{-4}$ with SEM). Left: heat kernel signature at a single point (i.e. $K_t(x) = K_t(x,x)$ over time). Right: heat map values between antipodal points (i.e. $K_t(x,y)$ where x,y are antipodal). Note the x-axis (time) is on a log-scale.*

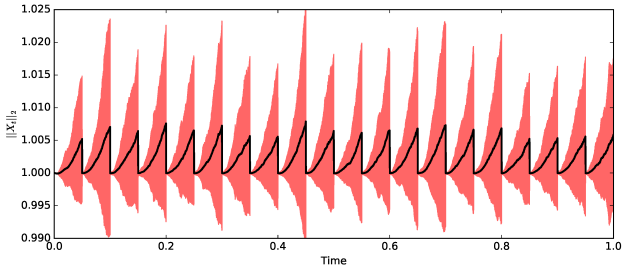

Figure 3: *Off-manifold distance of a diffusion process on the sphere, with the same settings as Figure 1. Black line represents the distance $||X_t||_2$, averaged over 500 trajectories for $t \in [0,1]$. Red shading is the standard deviation. Notice the spikes occurring every $T_\delta = 0.05$, which is due to the projection step (see section 3.3) moving the process back to the manifold.*

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 $||X_t||_2$ over time, where adherence to the sphere would imply $||X_t||_2 = 1$. The plot demonstrates the importance of T_δ, as the deviation from the correct distance continually 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, RBG*09], 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 $h_s = 0.01$, $T_\delta = 0.1$, $n_T = 500$, $\tau = \infty$, and $T_{\max} = 10^3$ 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 $k = 15$, $T_\delta = 0.25$, $h_s = 0.05$, $n_T = 200$, $T_{\max} = 500.0$, and $\eta_h = 14.9$, 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_t (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.

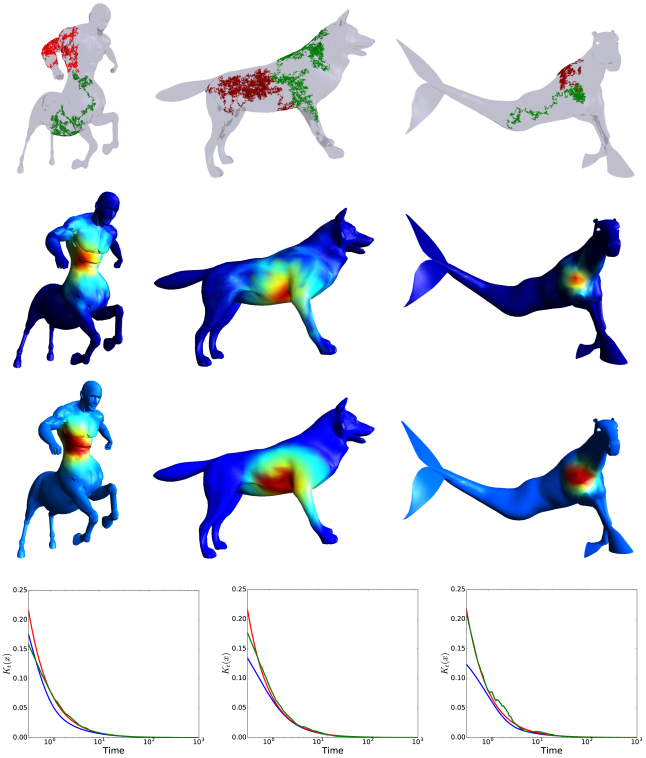

Figure 4: *First row: examples of Brownian motions on mesh manifolds. Second row: colours denote the log time-of-arrival ($\min\{t | K_t(x,y) > 10^{-4}\}$ for fixed x). Third row: same as second, but computed with linear finite element spectral approach instead. Fourth row: $K_t(x)$, with time on a log scale (blue: spectral, green: stochastic, red: short-time approximation via eq. 2). Note that the diffusion source point x is the same across columns.*

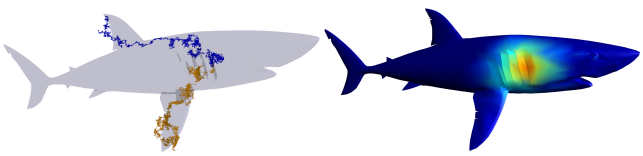

Figure 5: *Left: stochastic trajectories straying the from manifold surface. Right: stochastic time-of-arrival map (as in Figure 4).*

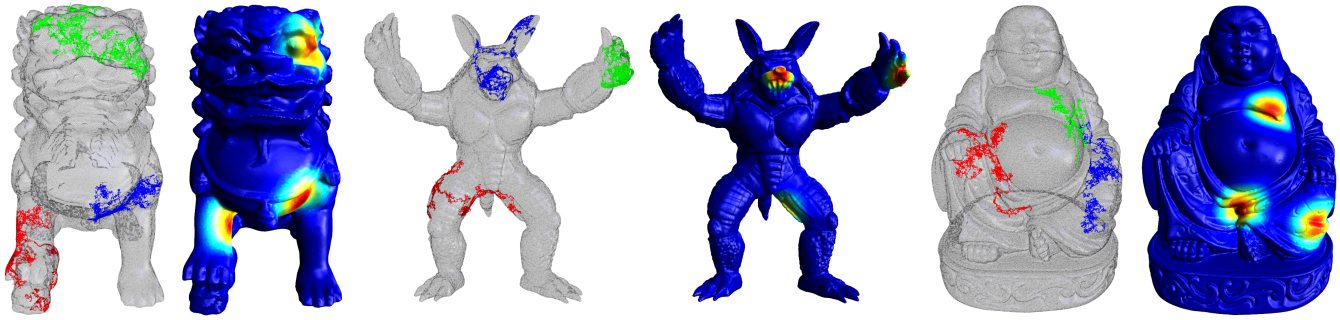

Figure 6: *Stochastic trajectories on models and colours as heat kernel sums $C(p) = \sum_i \sum_{p \in P} K_t(p, q_i)$ at $t = 25.0$, from three points q_i.*

Model	N	MLS (s)	$\lvert S \rvert$	STG (s)	KDE (s)
Chinese Lion	152807	30.5	1	159.31	38.75
			5	811.26	192.93
Armadillo	172974	36.75	1	174.02	43.24
			5	874.21	214.85
Buddha	719560	143.44	1	176.11	173.63
			5	896.88	870.8

Table 1: *Columns: model size N, number of feature points $|S|$, and computation time for MLS surface construction (including KD-tree), stochastic trajectory generation (STG), and KDE, respectively, in seconds. Experiments were performed on a Windows 8 machine (3.6GHz CPU and 16GB RAM) in a single thread.*

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, periodically projected onto MLS surfaces. We then developed a method for estimating the heat kernel on the manifold using sample trajectories from this process.

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 eigenvalue problem) and of working locally, mostly in the lower dimension of the submanifold. This suggests that our approach may be useful for manifolds of high co-dimension or in cases with an enormous number of samples, since our algorithm only requires a small set of nearby points at any given time. Our method is also easily parallelizable, since the trajectories are generated independently. The cost and stability of our algorithm does not depend on the time of interest t, as long as h_s is sufficiently smaller than t and $t < T_{\max}$, though larger times will be more expensive for trajectory generation (depending linearly on $T_{\max}$). Our algorithm allows the trade-off between accuracy and computation time to be controlled by altering its parameters, particularly n_T, and it has a less than quadratic dependence on N, suggesting its potential utility for large datasets. For the implementation considered here, computing the heat kernels of $|S|$ feature points costs $\sim O(N[|S| + \log(N)])$, excluding the dependence on other variables.

The computational complexity of the method of [PS13] is between $O(|S|N)$ and $O(|S|N^2)$, depending on the sparsity of the matrices involved. A qualitative sense of our performance against the method of [VBCG10] can be gained by comparing timing results on the models in Table 1. However, [VBCG10] computes the diagonal 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 improving our algorithm. For example, better local surface construction methods, such as those for the estimation of surface normals, are likely to be useful. The KDE-based transition density estimation approach is also amenable to improvement (e.g. via importance sampling [DG02] and forward-reverse representations [MSS*04]), as well as the bandwidth calibration. Further, it would be desirable to explore methods more capable of constraining the random process to the manifold surface. One possibility is to move the process on a surface constructed by interpolating several nearby local surfaces, thus avoiding the need for periodic projections. Separately, having time-adaptive T_δ and h_s could improve computational efficiency.

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 computation time. Further theoretical convergence analysis, with respect to manifold sampling density, integration step-size h_s, jump time T_δ, and other parameters, of the accuracy of both the SDE trajectories and the transition density, would be desirable as well.

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

[Avr03] AVRIEL M.: *Nonlinear Programming: Analysis and Methods*. Courier Corporation, 2003. 4

[BBK07] BRONSTEIN A. M., BRONSTEIN M., KIMMEL R.: Calculus of nonrigid surfaces for geometry and texture manipulation. *IEEE Transactions on Visualization and Computer Graphics 13*, 5 (2007), 902–913. 6

[BK10] BRONSTEIN M. M., KOKKINOS I.: Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (2010), IEEE, pp. 1704–1711. 1

[BN03] BELKIN M., NIYOGI P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation 15*, 6 (2003), 1373–1396. 3

[Bra03] BRAND M.: Charting a manifold. *Advances in neural information processing systems* (2003), 985–992. 3

[BSW08] BELKIN M., SUN J., WANG Y.: Discrete Laplace operator on meshed surfaces. In *Proceedings of the Twenty-Fourth Annual Symposium on Computational Geometry* (2008), ACM, pp. 278–287. 1

[BSW09] BELKIN M., SUN J., WANG Y.: Constructing Laplace operator from point clouds in Rd. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms* (2009), Society for Industrial and Applied Mathematics, pp. 1031–1040. 1

[BT96] BALLY V., TALAY D.: The law of the Euler scheme for stochastic differential equations: II. convergence rate of the density. *Monte Carlo Methods and Applications 2*, 2 (1996), 93–128. 3

[Chu06] CHUNG M. K.: Heat kernel smoothing on unit sphere. In *Biomedical Imaging: Nano to Macro, 2006. 3rd IEEE International Symposium on* (2006), IEEE, pp. 992–995. 5

[CLL*05] COIFMAN R. R., LAFON S., LEE A. B., MAGGIONI M., NADLER B., WARNER F., ZUCKER S. W.: Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences of the United States of America 102*, 21 (2005), 7426–7431. 1

[DG02] DURHAM G. B., GALLANT A. R.: Numerical techniques for maximum likelihood estimation of continuous-time diffusion processes. *Journal of Business & Economic Statistics 20*, 3 (2002), 297–338. 3, 7

[DLL*10] DEY T. K., LI K., LUO C., RANJAN P., SAFA I., WANG Y.: Persistent heat signature for pose-oblivious matching of incomplete models. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 1545–1554. 1

[GBAL09] GEBAL K., BÆRENTZEN J., AANÆS H., LARSEN R.: Shape analysis using the auto diffusion function. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 1405–1413. 2

[Hsu02] HSU E. P.: *Stochastic Analysis on Manifolds*, vol. 38. American Mathematical Soc., 2002. 2

[HW*96] HU Y., WATANABE S., ET AL.: Donsker's delta functions and approximation of heat kernels by the time discretization methods. *Journal of Mathematics of Kyoto University 36*, 3 (1996), 499–518. 3

[Itô62] ITÔ K.: The Brownian motion and tensor fields on Riemannian manifold. *Proc. Int. Congr. Math., Stockholm* (1962). 2

[IW14] IKEDA N., WATANABE S.: *Stochastic Differential Equations and Diffusion Processes*, vol. 24. Elsevier, 2014. 2

[JMS08] JONES P. W., MAGGIONI M., SCHUL R.: Manifold parametrizations by eigenfunctions of the Laplacian and heat kernels. *Proceedings of the National Academy of Sciences 105*, 6 (2008), 1803–1808. 1

[KH*97] KOHATSU-HIGA A., ET AL.: High order Itô-Taylor approximations to heat kernels. *Journal of Mathematics of Kyoto University 37*, 1 (1997), 129–150. 3

[LLWZ12] LIANG J., LAI R., WONG T. W., ZHAO H.: Geometric understanding of point clouds using Laplace-Beltrami operator. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (2012), IEEE, pp. 214–221. 2, 4

[LPG12] LIU Y., PRABHAKARAN B., GUO X.: Point-based manifold harmonics. *IEEE Transactions on Visualization and Computer Graphics 18*, 10 (2012), 1693–1703. 1

[LS81] LANCASTER P., SALKAUSKAS K.: Surfaces generated by moving least squares methods. *Mathematics of computation 37*, 155 (1981), 141–158. 3

[LZ13] LIANG J., ZHAO H.: Solving partial differential equations on point clouds. *SIAM Journal on Scientific Computing 35*, 3 (2013), A1461–A1486. 2, 4

[Mar55] MARUYAMA G.: Continuous Markov processes and stochastic equations. *Rendiconti del Circolo Matematico di Palermo 4*, 1 (1955), 48–90. 4

[Mat17] MATTHEW A.: *Critical fluctuations and coupling of stochastic neural mass models*. PhD thesis, The University of Queensland, 1 2017. 5

[MS67] MCKEAN H. P., SINGER I. M.: Curvature and the eigenvalues of the Laplacian. *J. Differential Geometry 1*, 1 (1967), 43–69. 2

[MSS*04] MILSTEIN G. N., SCHOENMAKERS J. G., SPOKOINY V., ET AL.: Transition density estimation for stochastic differential equations via forward-reverse representations. *Bernoulli 10*, 2 (2004), 281–312. 3, 4, 5, 7

[OG09] OZAKIN A., GRAY A. G.: Submanifold density estimation. In *Advances in Neural Information Processing Systems* (2009), pp. 1375–1382. 3, 5

[Oli07] OLIPHANT T. E.: Python for scientific computing. *Computing in Science & Engineering 9*, 3 (2007). 5

[OMMG10] OVSJANIKOV M., MÉRIGOT Q., MÉMOLI F., GUIBAS L.: One point isometric matching with the heat kernel. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 1555–1564. 1

[PS13] PATANÉ G., SPAGNUOLO M.: Heat diffusion kernel and distance on surface meshes and point sets. *Computers & Graphics 37*, 6 (2013), 676–686. 1, 7

[PVG*11] PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V., ET AL.: Scikit-learn: machine learning in python. *Journal of Machine Learning Research 12*, Oct (2011), 2825–2830. 5

[RBG*09] REUTER M., BIASOTTI S., GIORGI D., PATANÈ G., SPAGNUOLO M.: Discrete Laplace–Beltrami operators for shape analysis and segmentation. *Computers & Graphics 33*, 3 (2009), 381–390. 1, 6

[Ros97] ROSENBERG S.: *The Laplacian on a Riemannian Manifold: an Introduction to Analysis on Manifolds*. No. 31. Cambridge University Press, 1997. 2

[Röß10] RÖSSLER A.: Runge–Kutta methods for the strong approximation of solutions of stochastic differential equations. *SIAM Journal on Numerical Analysis 48*, 3 (2010), 922–952. 4

[RWP06] REUTER M., WOLTER F.-E., PEINECKE N.: Laplace–Beltrami spectra as "shape-dna" of surfaces and solids. *Computer-Aided Design 38*, 4 (2006), 342–366. 1, 6

[SOG09] SUN J., OVSJANIKOV M., GUIBAS L.: A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum* (2009), vol. 28, Wiley Online Library, pp. 1383–1392. 1, 2

[U*87] UEMURA H., ET AL.: On a short time expansions of the fundamental solution of heat equations by the method of Wiener functional. *Journal of Mathematics of Kyoto University 27*, 3 (1987), 417–431. 2

[VBCG10] VAXMAN A., BEN-CHEN M., GOTSMAN C.: A multi-resolution approach to heat kernels on discrete surfaces. *ACM Transactions on Graphics (TOG) 29*, 4 (2010), 121. 1, 4, 7