Noname manuscript No.
(will be inserted by the editor)

Multi-scale anisotropic heat diffusion based on normal-driven shape representation

Shengfa Wang · Tingbo Hou · Zhixun Su · Hong Qin

Received: date / Accepted: date

Abstract Multi-scale geometric processing has been a popular and powerful tool in graphics, which typically employs isotropic diffusion over scales. This paper proposes a novel method of multi-scale anisotropic heat diffusion on a manifold, based on a new normal-driven shape representation and edge-weighted heat kernels (EHK). The new shape representation, named as Normal-Controlled Coordinates (NCC), can encode local geometric details of a vertex along its normal direction and rapidly reconstruct the surface geometry. Moreover, the inner product of the NCC and its corresponding vertex normal, called Normal Signature (NS), defines a scalar/heat field over curved surface. The anisotropic heat diffusion is conducted using the weighted heat kernel convolution governed by local geometry. The convolution is computed iteratively based on the semigroup property of heat kernels towards accelerated performance. This diffusion is an efficient multi-scale procedure that rigorously conserves the total heat. We apply our new method to multi-scale feature detection, scalar field smoothing and mesh denoising, and hierarchical shape decomposition. We conduct various experiments to demonstrate the effectiveness of our method. The proposed method can be generalized to handle any scalar field defined over manifold.

Keywords Normal-Controlled Coordinates · Normal signature · Anisotropic diffusion · Edge-weighted heat kernel · Multi-scale feature extraction

S. Wang · Z. Su
Dalian University of Technology, 116024, China
E-mail: shengfawang@gmail.com

S. Wang · T. Hou · H. Qin
Stony Brook University (SUNY), NY 11794-4400, USA

1 Introduction

Recently, heat kernels and their utilities in diffusion start to gain momentum for geometric information processing in graphics, with applications in various research problems, including multi-scale feature detection [31], scalar field smoothing [26], automatic diffusion [8], shape matching [25], shape retrieval [2], etc. A typical diffusion process is conducted by convoluting an initial scalar/heat field with heat kernels. The advantage of such process is that, both diffusion and its kernel function afford robust multi-scale processing on manifolds, with intrinsic property of isometric-invariance. While these techniques have shown promise in multi-scale shape analysis, there still remain certain limitations in the current state-of-the-art, including initial heat field design, anisotropic diffusion, short-time scale behavior, improved performance, etc.

First, existing work oftentimes emphasizes the comprehensive studies of kernel functions and their properties, while paying far less attention to the initial heat field design. The initial field is a scalar function defined on the surface at time $t = 0$. It will gradually diffuse on the surface along t, resulting in different scales. The initial field is frequently assigned by using some simple characteristics such as curvature [20], texture [13], or other surface measurements [32,28]. These characteristics lack sufficient information to describe and reconstruct the shape. Moreover, they are sensitive to scale changes, which goes against the scale-invariant nature of the diffusion. To better depict the characteristics of a surface, an informative and stable initial field is strongly desirable. Second, all the previous heat diffusions are isotropic in nature, which are based on isotropic heat kernels on manifolds. Yet, anisotropic diffusion is much more desirable in many cases, such as smoothing and feature finding on surfaces with sharp edges. Anisotropic diffusion, which is much more powerful than standard isotropic diffusion, could control the diffusion direction by assigning weighted heat kernels to vertices. Third, while the kernel function itself can somehow represents surface geometry, it is hard to capture details in the short-time scale. Short-time heat kernels are intrinsically linked to high frequency spectrum of the shape, which are computation-wised costly to obtain. Therefore, heat kernels often start from a moderate scale and ignores short-time scales. To tackle this problem, an initial heat design that can capture the short-time details is much more favorable, and the initial scalar field with such attractive property can be naturally fed into the diffusion process that depicts the long-time behavior. In addition, the computation of heat diffusion may involve convolution over the entire surface, which could be extremely expensive. It must be accelerated to afford efficient multi-scale processing.

In this paper, our efforts are dedicated to address the aforementioned difficulties. We propose a normal-driven shape representation, called Normal-Controlled Coordinates (NCC), which have several outstanding properties, including being parallel with normal directions, scale-invariance, and properly behaving for open surfaces without tangential tension (Fig. 3 (right)), etc. Essentially, NCC can be viewed as an improved generalization of differential coordinates [21]. They can encode local details for each vertex and reconstruct mesh geometry. The inner product of NCC and its corresponding normal, defined as the Normal Signature (NS), is both rotation-invariant and scale-invariant, which is used to initialize a heat field in our work. After the field design, we devise a novel anisotropic heat diffusion based on edge-weighted heat kernels (EHK) directly derived from the NCC. The EHK can well control the anisotropic heat diffusion on surface, since such filter is properly weighted by local geometric information. Moreover, the EHK is stable to irregular sampling of surfaces by using local area as one weighting factor and can be transformed into a scale-invariant filter. In addition, the anisotropic diffusion field is a heat conservation field, which has clear physical meanings and avoids numerical degeneracy. The initial heat field coupled with its anisotropic diffusion, can naturally characterize and bridge both the short-time (local) and long-time (global) geometric behaviors. Inspired by the semigroup property of heat kernels, we use a multi-step method to diffuse the heat instead of computing heat kernels directly over a larger region for a longer time span. The heat diffusion is then reduced to simple matrix-vector multiplication. Since only sparse matrices are used, the diffusion process is very fast. Fig. 1 illustrates the pipeline of our approach. Specifically, the contributions of this paper are summarized as follows:

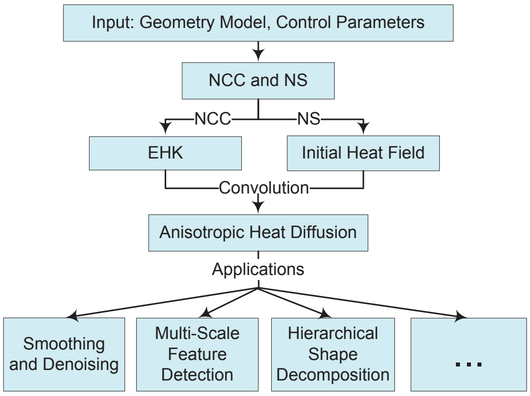

Input: Geometry Model, Control Parameters
NCC and NS
NCC
NS
EHK
Initial Heat Field
Convolution
Anisotropic Heat Diffusion
Applications
Smoothing and Denoising
Multi-Scale Feature Detection
Hierarchical Shape Decomposition
...

Fig. 1 The pipeline of our approach. We use NCC and NS to compute the EHK and the initial heat field, respectively. The diffusion process affords listed applications and could have more.

- We propose a normal-driven shape representation (NCC) towards feature definition. The inner product of NCC and the corresponding vertex normal (i.e. NS) is defined as an initial field for diffusion.
- We develop a novel method for anisotropic diffusion on manifolds, with anisotropic kernels (EHK) derived directly from the NCC.
- We conduct the diffusion using kernel convolution within a local region. An iterative method is utilized to diffuse the heat over the surface efficiently.
- We concentrate on three applications of our method, including multi-scale feature detection, scalar field smoothing and mesh denoising, and hierarchical model decomposition.

2 Related work

As a local shape descriptor, differential coordinates [1, 21] have been widely used for mesh processing during recent years because of their robustness, speed, and ease of implementation. Their application scopes also expand to industrial and artistic design. Nonetheless, the definition of previously-used differential coordinates appears to be incomplete on boundary vertices, and tangential tension may occur on boundaries of open surfaces (Fig. 3 (left)). Also the differential coordinates do not explicitly unite with their vertex normals, until the vertex normals are extraordinarily defined using cotangent weight [4]. Other descriptors, as proposed in the literature [24,29,22], are simple and efficient. However, they suffer from the problems of mesh quality and sampling, and are highly sensitive on scale changes. Local descriptors constructed by shape spectrum, such as the eigenfunctions of Laplace-Beltrami operator [8] and heat kernels [31], represent the shape information in a multi-scale way. While the global information can be well captured by the first few eigenfunctions, for local information, much more eigenfunctions are needed, which is computationally expensive. Furthermore, since the spectrum are globally defined, they are sensitive to boundary and topology changes.

Given an initial scalar field defined on the surface, Gaussian kernel was first adopted to conduct the diffusion [20,14,34,13], for its easy implementation. It was used to construct a multi-scale representation and then find features as local extrema. The Gaussian kernel can mimic the diffusion kernel in small regions on curved surfaces, but are not suitable for large regions. Ricci flow [35] has also been employed to construct a scale space by shape diffusion. While this diffusion intrinsically follows the curvature evolution, it diffuses the shape, not the scalar field. Recently, Patané and Falcidieno [26] utilized heat kernels spanned on Laplace-Beltrami eigenfunctions to smooth a scalar field on the surface. Heat kernels are the fundamental solutions to the diffusion equation on manifold, thus, they can correctly describe the diffusion at all scales. Because of the multi-scale property of heat kernels, their methods can smooth the scalar field in both local and global scales. Similar work can be found in [28]. However, computing the entire eigenfunctions of the Laplace-Beltrami operator necessary for short-time scales, is computationally expensive. Their methods used approximations of heat kernels, and thus may lose the property of heat conservation, which may result in degeneracy. Also, they have less control of heat diffusion on surface, since the heat kernels refer to insufficient geometry information, such as normals.

Anisotropic geometry diffusion and bilateral filter have been proposed to smooth bivariate data or general discretized surfaces. The anisotropic geometry diffusion [5,3,11] are either complex or sensitive to boundary and scale changes. The bilateral filter [18,6,30] focuses on only local geometry information, which may also suffer from the problems of mesh quality and sampling. Moreover, they may result in degeneracies due to energy loss during the diffusion procedure.

3 Normal-Controlled Coordinates (NCC)

Given a 2D manifold M, let (V, E) be vertex set V and edge set E that comprise an irregular triangular mesh of M. Without loss of generality, we assume that each vertex has at least three neighbors. If not, we simply subdivide the corresponding triangulation by adding a vertex in the centroid.

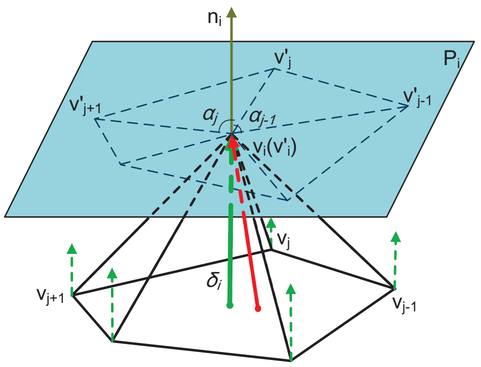


Fig. 2 The construction of NCC (colored in green) at a vertex. Tangential components on the projection plane and normal components are also highlighted. The conventional differential coordinates (without normal-control) of this vertex is also shown (colored in red), which is not parallel to the vertex normal.

The NCC at vertex v_i is defined as

$$\delta_i = \mathcal{N}(v_i) = \sum_{j \in N(i)} \omega_{ij}(v_i - v_j), \tag{1}$$

where $\mathcal{N}$ denotes the NCC operator, $N(i)$ is the one-ring neighborhood of vertex v_i, and ω_{ij} is the coefficient of the NCC operator, also called NCC weight. The inner product of NCC and the corresponding vertex normal is defined as normal signature (NS). It effectively represents local geometry information, such as feature size, convexity, and concavity.

The global geometry can then be decomposed into two sets of scalar data representing tangential and normal components, which collectively encode local parameterization (along local tangent plane) and local geometry information (perpendicular to local tangent plane), respectively. The local parameterization information is captured by the coefficients of the NCC operator $\{\omega_{ij}\}$, and the local geometry information is captured by the NS. Given any kind of evaluations on vertex normals, the local parameterization and geometry information are uniquely defined. Specifically, we use the mean weighted by areas of adjacent triangles (MWAAT) normals [17] in our experiments. Oftentimes, the local parameterization information can be considered to be unchanged, and hence only the operation of normal components is necessary. And the mesh geometry can be recovered linearly from coefficients of the NCC operators and the updated NS with the aid of vertex normals. In the following section, we will provide the derivation of NCC.

3.1 Derivation of NCC

For each vertex v_i, let $n_i = (n_{ix}, n_{iy}, n_{iz})$ be the vertex normal, and P_i be its projection plane that can be any plane with the normal n_i. We formulate P_i as the tangent plane of v_i,

$$P_i = n_{ix}x + n_{iy}y + n_{iz}z - n_i \cdot v_i = 0. \tag{2}$$

We define the normal weight $\widetilde{\omega}_{ij}$ of the neighbor vertex v_j as,

$$\widetilde{\omega}_{ij} = \frac{\tan(\alpha_j/2) + \tan(\alpha_{j-1}/2)}{\|v_i' - v_j'\|}, \tag{3}$$

where v_j' denotes the projection of v_j onto the plane P_i, and angles α_j and α_{j-1} are computed from the projection plane (as shown in Fig. 2). Note that the normal weight of v_j indirectly relates to v_j via projection. It is the mean value coordinate weight [19] on the projection plane.

The weight in Eq. (1) is the normalized normal weight $\omega_{ij} = \widetilde{\omega}_{ij} / \sum_{j \in N(i)} \widetilde{\omega}_{ij}$. To make NCC insensitive to sampling and scale, we rescale the NCC by areas, and Eq. (1) becomes

$$\delta_i' = \delta_i / a_i^{1/2} = \sum_{j \in N(i)} \omega_{ij}'(v_i - v_j), \tag{4}$$

where a_i is the Voronoi area centered at v_i [24], and $\omega_{ij}' = \omega_{ij}/a_i^{1/2}$. Note that, all the NCC mentioned later are rescaled by Voronoi areas.

Assembling Eq. (4) at each vertex, we obtain a linear system:

$$\mathbf{NV} = \boldsymbol{\delta} = \mathbf{S}\mathbf{n}_v, \tag{5}$$

where $\boldsymbol{\delta}$ is a vector consisting of NCC, $\mathbf{S}$ is a matrix consisting of NS, $\mathbf{n}_v$ is a vector consisting of vertex normals, and $\mathbf{N}$ is a sparse matrix with following elements

$$N_{ij} = \begin{cases} 1/a_i^{1/2}, & i = j, \\ -\omega_{ij}', & (i,j) \in E, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Eq. (5) can be effectively used to edit and reconstruct meshes in our applications.

3.2 Properties of NCC

NCC are generalized from differential coordinates in principle. They share some common properties of conventional differential coordinates, such as easy implementation, efficiency, etc. Moreover, NCC have their distinctive properties:

- NCC are always parallel with the corresponding vertex normals. The detailed proof is documented in the appendix.

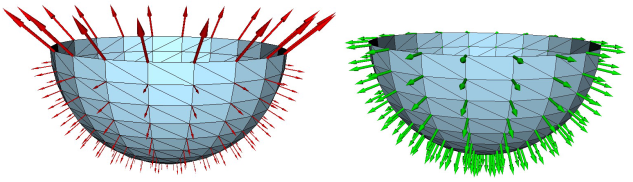


Fig. 3 Comparison between NCC and Laplacian coordinates. Laplacian coordinates have tangential tension on the boundary (colored in red), while NCC are well-defined on the boundary and insensitive to the sampling (colored in green).

- NCC are scale-invariant. Assume that we scale the model M by s, then the NCC of sM are given by

$$\begin{aligned} \delta_i(sM) &= \sum_{j \in N(i)} \frac{\omega_{ij}}{(s^2 a_i)^{1/2}}(sv_i - sv_j) \\ &= \sum_{j \in N(i)} \frac{\omega_{ij}}{(a_i)^{1/2}}(v_i - v_j) = \delta_i(M). \end{aligned} \tag{7}$$

- NCC are well defined on boundary. For open meshes, the normals of boundary vertices can be well defined. Fig. 3 shows the Laplacian coordinates and NCC. The tangential tension appears in the Laplacian coordinates (a), while it completely disappears in NCC (b).

These properties are significant for mesh processing. They afford the stability, efficiency, and a wide range of applications of NCC. In our work, we use the NS to initialize a scalar field as the heat at $t = 0$ for diffusion.

4 Edge-weighted heat kernels (EHK)

It is well-known that the heat diffusion over a manifold M is governed by the heat equation

$$\begin{cases} \frac{\partial u(x,t)}{\partial t} = -\Delta u(x,t), & t \in R^+, \\ u(x,0) = f(x), & x \in M, \end{cases} \tag{8}$$

where Δ is the Laplace-Beltrami operator, $f(v)$ is the initial heat defined on M. The results of Eq. (8) can be obtained by

$$u(x,t) = \int_M h_t(x,y) f(y) dy, \tag{9}$$

where $h_t(x,y)$ is the heat kernel between x and y at time t.

Heat kernels have many nice properties [31]. For instance, they are symmetric: $h_t(x,y) = h_t(y,x)$, and satisfy the semigroup identity: $h_{t+s}(x,y) = \int_M h_t(x,z) h_s(y,z) dz$. They are also isometric-invariant, multi-scale, informative, and stable. However, the commonly-used heat kernels do not consider some key geometry information, such as normals and sampling of meshes. As a result, they lack control of heat diffusion and are sensitive to irregular sampling. A weighted linear finite element method (FEM) discretization of heat kernels was given in [26]. They discretized the Laplace-Beltrami operator in Eq. (8) by adding area weights. This discretization is stable to the sampling with the cost of losing symmetry. Despite of the improvements, they still lack control of the direction of heat diffusion.

At each vertex, heat diffuses through its connected paths (edges) as time goes by. Therefore, the edge weight plays an important role in thermal conductivity. A proper edge weight that can control the heat diffusion is preferred. Moreover, the area weights are used to compute eigen-system only to combat the problem of irregular sampling and symmetry of heat kernels. The Laplace-Beltrami operator is discretized by the edge-weighted Laplacian matrix $\tilde{\mathbf{L}} := \mathbf{B}^{-1}\mathbf{L}^e$ [26,27],

$$\mathbf{L}^e(i,j) = \begin{cases} w(i,j) := e^{-\frac{\|\delta_i - \delta_j\|_2^2}{\sigma^2}}, & j \in N(i), \\ -\Sigma_{k\in N(i)} w(i,k), & i = j, \\ 0, & \text{otherwise}, \end{cases} \tag{10}$$

$$\mathbf{B}(i,j) = \begin{cases} \frac{|t_r|+|t_s|}{12}, & j \in N(i), \\ \frac{\Sigma_{k\in N(i)}|t_k|}{6}, & i = j, \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

where σ is a parameter proportional to the variance of NCC, t_r and t_s are the triangles that share the edge (i,j), and $|t_r|$ is the area of triangle t_r. Here, $\mathbf{B}$ is a mass matrix used to compensate the irregular sampling, and $\mathbf{L}^e$ is a weighted matrix controls the tendency of heat diffusion (e.g. isotropic or anisotropic) using parameter σ, which will be further discussed in Section 6.

The generalized eigen-system $\{(\lambda_i, \boldsymbol{\phi}_i)\}_{i=1}^n$ of $(\mathbf{L}^e, \mathbf{B})$ satisfies

$$\mathbf{L}^e \boldsymbol{\phi}_i = \lambda_i \mathbf{B} \boldsymbol{\phi}_i, \quad i = 1, 2, \ldots, n. \tag{12}$$

Using the eigenfunctions, EHK can be analytically written as

$$h_t^e(i,j) = \sum_{k=1}^{n} e^{-\lambda_k t} \boldsymbol{\phi}_k(i) \boldsymbol{\phi}_k(j). \tag{13}$$

The new heat kernels are determined by the discretized edge-weighted Laplace-Beltrami operator, which incorporates more geometry. We can control the heat diffusion easily by adjusting the edge weight $w(i,j)$. Intuitively, the heat diffuses faster along the prominent parts, but rather slow when cutting across them, such as sharp edges. Fig. 4 (a) and (b) show the difference between commonly-used heat kernels and our new EHK, and it clearly illustrates that EHK are more aware of local geometry. Moreover, EHK are robust w.r.t. noise and perturbation (Fig. 4 (c)).

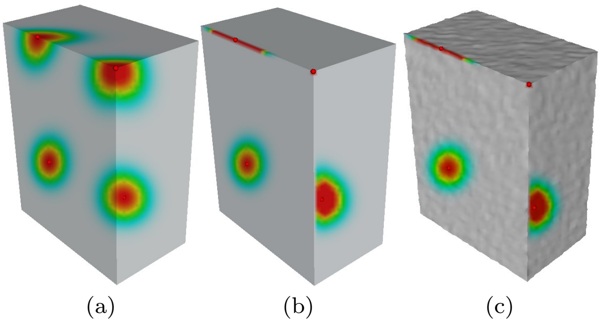


Fig. 4 Comparison between the commonly-used heat kernels (a), our EHK (b) and EHK on the noisy surface (c). The heat kernels of four different vertices (highlighted as small red balls) on a cuboid are shown, respectively.

The discretization of the heat kernel introduced above can be easily refined to be scale invariant using the method in [26], if scale changes are encountered. Specifically, assume a manifold M is scaled to sM. The matrix $\mathbf{L}^e$ is unchanged, the mass matrix $\mathbf{B}$ becomes $s^2\mathbf{B}$, and the eigen-system turns into $\{(\lambda_i/s^2, \boldsymbol{\phi}_i/s)\}_{i=1}^n$. The EHK can be modified to

$$h'_t(i,j) = \beta \sum_{k=1}^{n} e^{-\lambda'_k t} \boldsymbol{\phi}_k(i) \boldsymbol{\phi}_k(j), \tag{14}$$

with $\beta = \frac{B(i,i)+B(j,j)}{2}$, and $\lambda'_k = \frac{\lambda_k}{\lambda_2}$, which is scale invariant.

5 Anisotropic diffusion by local convolution

Given an initial heat field f_0 defined by the NS of M, we can diffuse it isotropically or anisotropically using EHK. We exploit heat kernel convolution in a local region to ensure the heat conservation and accelerate the computation. Before examining the convolution, let us first investigate the relationship between diffusion region and diffusion time.

The multi-scale property of heat kernels implies that for small value of time t, heat kernels can be well approximated by the ones within a small geodesic neighborhood of vertex v [31,25]. An explicit relationship between time and the size of diffusion region was given in [9]. The heat kernel $h_t(v,\cdot)$ can be viewed as the transition density function of the Brownian motion, and is determined by the geodesic ball $B(v, O\sqrt{c_n t \log t})$. Mémoli [23] further justified the interpretation of time as a geometric scale from the view point of homogenization of partial differential equations (PDE). Therefore, it is justifiable and acceptable to define a heat region around v_i as

$$\Omega_t^i = \{v_j | h_t^e(v_i, v_j) > \tau(t)\}, \tag{15}$$

where $\tau(t)$ is a threshold, which is set to $0.01/(1+t)$ according to our experience of the experiments. The heat region can be found efficiently by taking advantage of the nearest neighbor method [10]. Moreover, the heat region of each vertex for a fixed time t only needs to be computed once. The region is invariant during the diffusion using our iterative local heat kernel convolution.

The updated heat field after one-time convolution in the heat region is

$$f(v_i, t) = \sum_{v_j \in \Omega_t^i} \tilde{h}_t(v_j, v_i) f_0(v_j) \tag{16}$$

where $\tilde{h}_t(v_j, v_i) = \frac{h_t^e(v_j, v_i)}{\sum_{v_k \in \Omega_t^j} h_t^e(v_j, v_k)}$, and $f_0(v_j)$ is the value of initial field at v_j. They can be written in matrix form

$$\mathbf{F}_t = \mathbf{A}_t \mathbf{F}_0, \tag{17}$$

where $\mathbf{F}_t = [f(v_1, t), \ldots, f(v_n, t)]^T, \mathbf{F}_0 = [f_0(v_1), \ldots, f_0(v_n)]^T$, and $\mathbf{A}_t$ is a sparse matrix with elements

$$\mathbf{A}_t(i, j) = \begin{cases} \tilde{h}_t(v_j, v_i), & \text{if } v_j \in \Omega_t^i \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

We fix time $t = t_0$ for one-step diffusion in local heat region. According to the semigroup identity of heat kernels, the heat diffusion can be obtained by using the local heat kernel convolution iteratively. The heat field after k steps (or equivalently, kt_0 time) is

$$\mathbf{F}^k = \mathbf{A}_{t_0} \mathbf{F}^{k-1} = \ldots \mathbf{A}_{t_0}^{k-1} \mathbf{F}^1 = \mathbf{A}_{t_0}^k \mathbf{F}_0, k = 0, 1, \ldots. \tag{19}$$

When $k = 0$, the heat field is the initial field. As k increases, the heat diffuses to a larger region, and more global behavior results from the convolution. The diffusion turns into matrix-vector multiplication. Since $\mathbf{A}_{t_0}$ is a sparse matrix, the computation is very fast.

6 Numerical computation and implementation

Approximation The eigenvalues and eigenfunctions in Eq. (12) are computed by solving the generalized eigenproblem. The discrete heat kernel is approximated by the first m smallest eigenvalues and the corresponding eigenfunctions, which contribute to the shape most. Eq. (13) can be approximated by

$$h_t^e(i, j) = \sum_{k=1}^{m} e^{-\lambda_k t} \boldsymbol{\phi}_k(i) \boldsymbol{\phi}_k(j),\ m < n. \tag{20}$$

As a remaining challenge for large data, the computational cost may be incredibly high. In this case, a multi-resolution approach can be used to accelerate the computation of heat kernels. Vaxman et al. [33] showed that heat kernels at any time can be properly approximated by a lower resolution version of the original surface. To address this problem, we use a hierarchical scale space based on a pyramid representation which consists of consecutive layers $\{M^0, M^1, ..., M^h\}$ [13], where h is the number of layers, and M^0 is the original mesh. There is a large literature on mesh simplification using various error metrics. General approaches such as Progressive mesh [12] and QSlim [7] suffice for this purpose. The EHK on M^0 can be approximated on the simplified mesh M^h by

$$\mathbf{H}_t^e(M^0) = \mathbf{P}_h^1 \mathbf{H}_t^e(M^h) \mathbf{P}_1^h, \tag{21}$$

where $\mathbf{H}_t^h$ consists of the EHK at the coarsest resolution level M^h, and $\mathbf{P}_h^1$, $\mathbf{P}_1^h$ are prolongation matrices using barycentric coordinates [33].

Parameters There are two important parameters σ and k in our approach. The parameter σ in Eq. (10) controls the tendency of heat diffusion in some sense. When the data has noise, σ should be larger, and vice versa. On the other hand, if the preservation of sharp features is important, σ should be small so that neighboring NCC deviating far from the current one make very small contributions to itself. The parameter k in Eq. (19) controls the diffusion level. In a nutshell, the bigger the k is, the larger region the heat will diffuse to. The other parameters, such as time t and the number of eigenvalues m, are robust in our approach. To make the time t meaningful and stable for different surfaces, we rescale the surface such that the total area is equal to the number of vertices. Thus, $t = 1$ results in an average influence region of about 1-ring size [33], i.e., one step in random walk. In our experiments, we set $\sigma = 0.1$ for the feature-preserving purpose, $\sigma = 1$ for the smooth diffusion purpose, and fix $t = 1$ and $m = 150$, if no special requirements are documented.

7 Applications and experimental results

In this section, we will first illustrate the heat conservation of our method, by showing the variability of total heat of a given heat field. Then, we detail applications of our method, including multi-scale feature detection, scalar field smoothing and mesh denoising, and hierarchical shape decomposition. All the experiments are conducted on a computer with 2.67GHz CPU and 4G RAM. Our system is implemented using C++ with invoked Matlab function ‘eigs’ to compute eigendecomposition. Most computational costs of our approach can be carried out in the pre-processing stage, such as the computation of NCC, solving eigen-system, finding local heat regions and the Cholesky decomposition of the matrix in our reconstruction system. Table 1 documents the statistics and time performance of key parts in our experiments, where both synthetic and scanned models are utilized.

Table 1 Time performance of experiments. The 3rd, 4th, and 5th columns show the timing (in seconds) for constructing NCC, solving eigen-problem, and computing local convolution (k=1) at all points.

Models	# Vertices	NCC	Eigen	Convolution
Blob	8036	0.829	4.798	0.153
Gargoyle	10002	1.053	5.889	0.196
Fandisk	6475	0.619	3.944	0.119
Dinosaur	28287	2.919	16.554	0.519

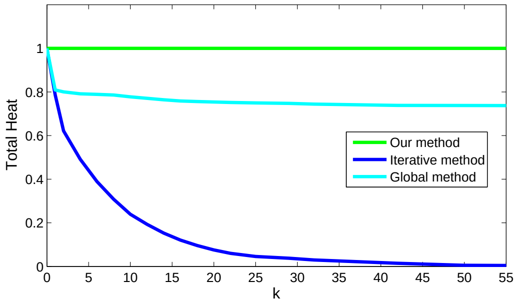
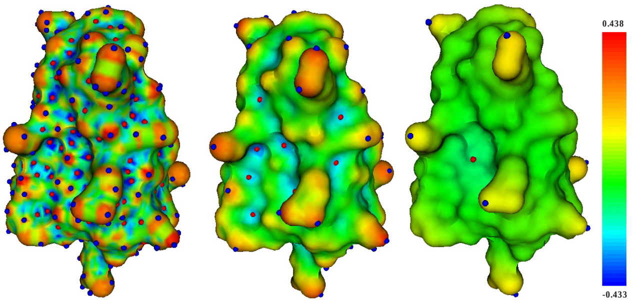


Fig. 5 The plot of total heat on the Fandisk with different methods. For convenience, we normalize the total heat to 1.

Heat conservation field A heat conservation field is constructed via iterative local heat kernel convolution. The total heat of a field defined on M at scale k can be measured by $E(M,k) = \sum_{i=1}^{n} \mathbf{F}^k(i)$, where $\mathbf{F}^k(i)$ is the value of scalar field $\mathbf{F}^k$ at vertex v_i. The global method [26] and the iterative method [28] can approximate and smooth a heat field, but do not conserve the total heat of the field, which might lead to degeneracy. It is easy to proof that our method rigorously ensures the heat conservation in the field, which is significant for applications (e.g. the last application proposed in this section). In Fig. 5, we illustrate the total heat at different scale compared with these methods. Obviously, our method conserve the total heat precisely, while the other two methods generate evident degeneracy. Furthermore, our approach is more efficient, since we convolve the heat kernels in the local region, which can be reused during the diffusion procedure. The computation complexity of field construction in those methods is $O(mn^2)$, while it is only $O(mnl)$ in our approach, where $l \ll n$ is the number of vertices in the local heat region.

Multi-scale feature detection Given a heat field constructed using local geometry quantities, such as NS, we can detect multi-scale features by analyzing the heat field and the heat diffusion. We declare a point v_i as a convex feature in scale k, if both of the following criteria are met:



(a)+: 208, -: 90 (b)+: 48, -: 22 (c)+: 10, -: 2

Fig. 6 Multi-scale feature detection on the Blob (initial scalar field is assigned by NS, $\sigma = 1$ and $c = 0.8$). (a) The features in scale $k = 0$. (b) The features in scale $k = 10$. (c) The features in scale $k = 40$. The convex (+) and concave (-) features are highlighted in blue and red balls, respectively.

1. $\mathbf{F}^k(i) > \mathbf{F}^k(j)$, j is the index of 2-ring neighborhood vertex of v_i.
2. $\mathbf{F}^k(i) > c \max\{\mathbf{F}^k(\cdot)\}$, $c \leq 1$ is a parameter.

The concave features can be defined in a similar way. For small k, the detected features depend on more local geometry information. For large k, more global information is taken into account. Fig. 6 shows the features detected in different scales. The features detected in the scale $k = 0$ are easily affected by the noise, and may be redundant. As k increases, the heat field becomes smooth gradually, and the features detected in different scales can well depict the shape information in a multi-scale sense. We also compare our method with the heat kernel signature (HKS) [31], as shown in Fig. 7. The HKS fails to distinguish convex and concave features, and has a limitation on small-scale features at short time even though all the eigenfunctions are used, while our method performs well in both cases. Moreover, our method can also find other types of features by setting the scalar field as different characteristics, such as texture, mass, density, conductivity, etc, as long as they are available.

Scalar field smoothing and mesh denoising One of the most direct applications of our approach is to smooth the scalar field defined on mesh surfaces. Given any scalar field without sharp features, we can smooth the scalar field using a large σ. In this case, our method and the isotropic heat kernel smoothing methods [26,28] perform almost the same in principle. Roughly speaking, the larger σ is, the smoother the scalar field will be. For the scalar field with sharp features, our approach smooths the scalar field using a small σ, and it preserves the sharp features better than the two aforementioned methods (Fig. 8 (a)-(c)). Moreover, our method has less

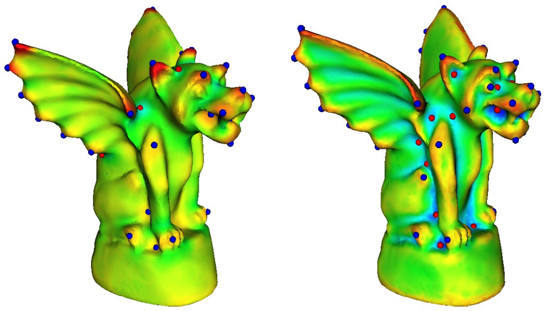


Fig. 7 Comparison of feature detection on the Gargoyle between the HKS (left) and our method (right, $\sigma = 0.1$ and $c = 0.8$). HKS has a limitation on small-scale features, such as features in the eyes, and fails to distinguish convex features (blue balls) and concave features (red balls), such as features inside mouth.

deviation from the initial field because of the conservation of total heat.

After obtaining a smooth scalar field assigned by NS, a smooth mesh can be reverse-engineered by updating the vertex positions. We compute the new vertex positions using NCC-based mesh reconstruction in a least squares sense. To better reconstruct the surface, we update the normals before updating the vertex positions by

$$n'_i = \frac{1}{\sum_{j \in N(i)} s_j} \sum_{j \in N(i)} s_j n_j, \tag{22}$$

where s_j is the NS at vertex v_j. The reconstruction system derived from the linear system in Eq. (5) is

$$\begin{bmatrix} \mathbf{N} \\ \boldsymbol{\omega}\mathbf{I} \end{bmatrix} \mathbf{V}' = \begin{bmatrix} \mathbf{S}'\mathbf{n}'_v \\ \boldsymbol{\omega} V \end{bmatrix}, \tag{23}$$

where $\mathbf{V}'$ is the matrix of unknown vertices, $\boldsymbol{\omega}$ is a weight matrix being set as the corresponding NS here, $\mathbf{S}'$ and $\mathbf{n}'_v$ are the updated NS and vertex normal, respectively. Because of the anisotropic diffusion on shape, our method performs well, especially for models with sharp feature. We compare our method with the heat kernel smoothing [28]. As shown in Fig. 8 (e)-(f), the sharp features are better persevered using our method.

Hierarchical decomposition and processing In addition, we examine a novel application of hierarchical decomposition of signals (scalar field) over 3D models. A scalar field defined on a 3D model can also be viewed as a 3D signal. Given any signal on a model $\mathbf{S} = \mathbf{F}_0$, we can decompose it into different frequency details and a residual base signal as

$$\mathbf{S} = \sum_{j=1}^{k} \mathbf{D}^j + \mathbf{S}^k, \ \mathbf{D}^j = \mathbf{F}^{j-1} - \mathbf{F}^j, \tag{24}$$

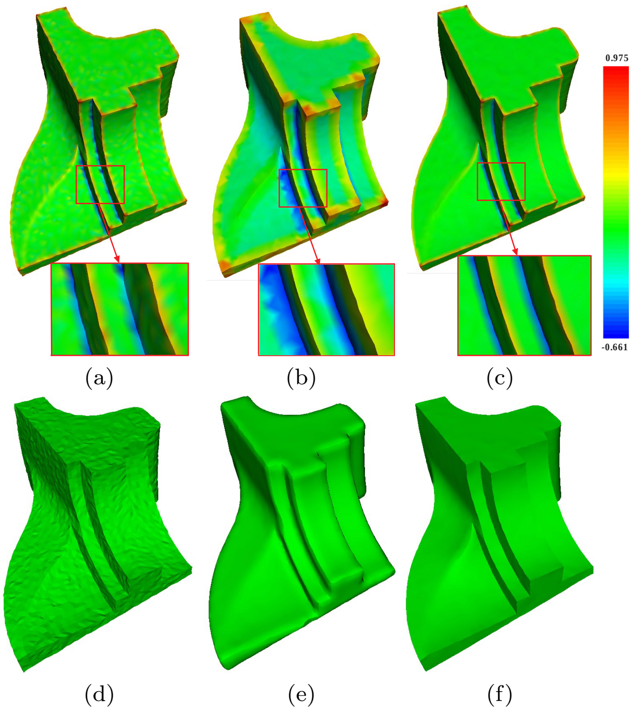


Fig. 8 Scalar field smoothing and mesh denoising. (a) Scalar field assigned by NS of Fandisk (Gaussian noise of 5% mean edge length). (b), (c) The smooth scalar fields computed by the method in [26] ($t = 4$) and our method ($\sigma = 0.1, k = 4$). (d) Fandisk mesh with Gaussian noise (5% mean edge length). (e), (f) The smooth meshes obtained by the method in [28] and our method.

where $\mathbf{D}^k$ is the detail in k-th level decomposition, $\mathbf{S}^k = \mathbf{F}^k$ is the residual base signal, and $\mathbf{F}^j$ is the heat field computed in Eq. (19). For small k, $\mathbf{D}^k$ represents the high frequency information of signal $\mathbf{S}$, and vice versa. Note that, one level decomposition here could contain several steps in the diffusion processing for the sake of efficiency.

Because of the conservation of total heat by our method, that is $\sum_{i=1}^{n} \mathbf{F}^k(i) = \sum_{i=1}^{n} \mathbf{F}^{k-1}(i)$, we can easily get

$$\sum_{i=1}^{n} \mathbf{D}^k(i) = 0, \tag{25}$$

where $\mathbf{D}^j(i)$ is the value of $\mathbf{D}^j$ at vertex v_i. Our decomposition in Eq. (24) is very similar to the empirical mode decomposition (EMD) [15,16], which is proposed so far for the function defined in the Euclidean space. In principle, it can be viewed as a new extension of the EMD to 3D models. In Fig. 9, we decompose a signal into 3 levels of different frequency details and a residual base signal respectively, that is $\mathbf{S} = \sum_{j=1}^{3} \mathbf{D}^j + \mathbf{S}^3$, and in each level decomposition, $\sum_{i=1}^{n} \mathbf{D}^j(i) = 0, 1 \leq j \leq 3$.

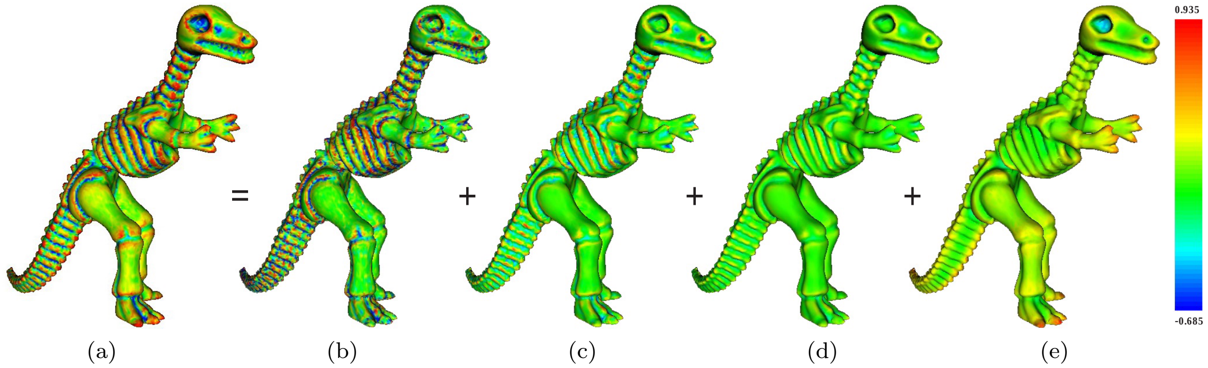


Fig. 9 Hierarchical decomposition of 3D signals (assigned by NS, $\sigma = 0.5$). (a) The signals on the Dinosaur. (b) The 1st level decomposition $\mathbf{D}^1$. (c) The 2nd level decomposition $\mathbf{D}^2$. (d) The 3rd level decomposition $\mathbf{D}^3$. (e) Residual base signal $\mathbf{S}^3$.

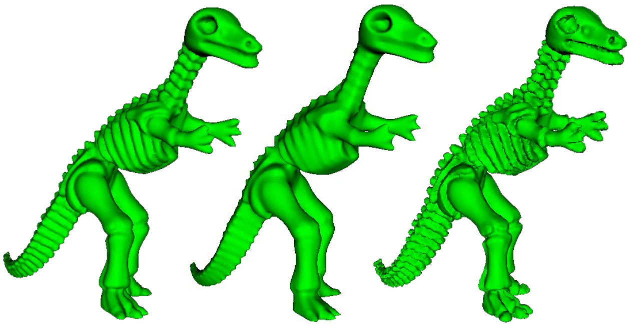


Fig. 10 Mesh reconstruction from 3D signals. (Left) Original Dinosaur model. (Center) Reconstruction from a smoothing filter ($w_1 = w_2 = 0$, $w_3 = 1$). (Right) Reconstruction from an enhancement filter ($w_1 = 1$, $w_2 = w_3 = 3$).

Furthermore, we can process the signal in different levels of detail

$$\tilde{\mathbf{S}} = \sum_{j=1}^{k} w_j \mathbf{D}^j + \mathbf{S}^k, \tag{26}$$

where w_j is the parameter controlling the weight of the j-th level detail. They can be edited conveniently according to different applications, such as enhancement filter, smoothing filter, watermarking, etc. Then, the updated shape M corresponding to the new signal $\tilde{\mathbf{S}}$ can be obtained using the reconstruction system in Eq. (23). Fig. 10 shows the results of mesh reconstruction from the updated 3D signals.

8 Conclusion

In this paper, we have studied both shape and scalar diffusion on curved surfaces, with novel solutions in many aspects, including NCC as local shape representation, anisotropic diffusion using EHK, and iterative local convolution to reduce the computational cost. The proposed solutions comprise a complete and versatile system for multi-scale processing, which is valuable in many applications with advantageous properties and improved performance. Several applications, such as multi-scale feature detection, scalar field smoothing and mesh denoising, and hierarchical model decomposition are pursued to showcase the broad utility of our approach.

In the near future, we plan to apply our method to define and detect different types of features, such as point features, line features, and area features. Surface segmentation and model watermarking are also some valuable applications that we plan to explore. Moreover, extending this approach to 3D volumetric datasets deserves further investigation which can broaden our method's application scopes.

Acknowledgements The authors would like to express their deepest gratitude to the anonymous reviewers for their extensive help in improving this article. This research is supported by the Fundamental Research Funds for the Central Universities, National Natural Science Foundation of China-Guangdong Joint Fund grant U0935004, National Natural Science Foundation of China 60873181, and National Science Foundation grants IIS-1047715, IIS-1049448, IIS-0949467 and IIS-0710819. Models are courtesy of the AIM@SHAPE repository.

References

1. Alexa, M.: Differential coordinates for local mesh morphing and deformation. The Visual Computer **19**(2), 105–114 (2003)
2. Bronstein, A.M., Bronstein, M.M., Guibas, L.J., Ovsjanikov, M.: Shape google: geometric words and expressions for invariant shape retrieval. ACM Trans. Graph. **30**(1), 1–20 (2010)
3. Clarenz, U., Diewald, U., Rumpf, M.: Anisotropic geometric diffusion in surface processing. In: Visualization, pp. 397–405 (2000)
4. Desbrun, M., Meyer, M., Schröder, P., Barr, A.H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In: SIGGRAPH, pp. 317–324 (1999)

5. Desbrun, M., Meyer, M., Schröder, P., Barr, A.H.: Anisotropic feature-preserving denoising of height fields and bivariate data. In: Graphics Interface, pp. 145–152 (2000)
6. Fleishman, S., Drori, I., Cohen-Or, D.: Bilateral mesh denoising. ACM Trans. Graph. **22**(3), 950–953 (2003)
7. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: SIGGRAPH, pp. 209–216 (1997)
8. Gębal, K., Bærentzen, J.A., Aanæs, H., Larsen, R.: Shape analysis using the auto diffusion function. In: Symposium on Geometry Processing, pp. 1405–1413 (2009)
9. Grigor'yan, A.: Escape rate of brownian motion on riemannian manifolds. Applicable Analysis **71**(1), 63–89 (1999)
10. Hastie, T., Tibshirani, R.: Discriminant adaptive nearest neighbor classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **18**(6), 607–616 (1996)
11. Hildebrandt, K., Polthier, K.: Anisotropic filtering of non-linear surface features. Computer Graphics Forum **23**(3), 391–400 (2004)
12. Hoppe, H.: Progressive meshes. In: SIGGRAPH, pp. 99–108 (1996)
13. Hou, T., Qin, H.: Efficient computation of scale-space features for deformable shape correspondences. In: ECCV, pp. 384–397 (2010)
14. Hua, J., Lai, Z., Dong, M., Gu, X., Qin, H.: Geodesic distance-weighted shape vector image diffusion. TVCG **14**(6), 1643–1650 (2008)
15. Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N.C., Tung, C.C., Liu, H.H.: The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **454**(1971), 903–995 (1998)
16. Huang, N.E., Wu, Z.: A review on hilbert-huang transform: Method and its applications to geophysical studies. Reviews of Geophysics **46**(2), RG2006+ (2008)
17. Jin, S.: A comparison of algorithms for vertex normal computation. The Visual Computer **21**(1-2), 71–82 (2005)
18. Jones, T.R., Durand, F., Desbrun, M.: Non-iterative, feature-preserving mesh smoothing. ACM Trans. Graph. **22**(3), 943–949 (2003)
19. Ju, T., Schaefer, S., Warren, J.: Mean value coordinates for closed triangular meshes. In: SIGGRAPH, pp. 561–566 (2005)
20. Lee, C.H., Varshney, A., Jacobs, D.W.: Mesh saliency. ACM Trans. Graph. **24**(3), 659–666 (2005)
21. Lipman, Y., Sorkine, O., Cohen-Or, D., Levin, D., Rössl, C., Seidel, H.P.: Differential coordinates for interactive mesh editing. In: SMI, pp. 181–190 (2004)
22. Lipman, Y., Sorkine, O., Levin, D., Cohen-Or, D.: Linear rotation-invariant coordinates for meshes. ACM Trans. Graph. **24**(3), 479–487 (2005)
23. Mémoli, F.: Spectral gromov-wasserstein distances for shape matching. In: NORDIA, pp. 256–263 (2009)
24. Meyer, M., Desbrun, M., Schrder, P., Barr, A.H.: Discrete differential-geometry operators for triangulated 2-manifolds **3**(7), 1–26 (2002)
25. Ovsjanikov, M., Mérigot, Q., Mémoli, F., Guibas, L.: One point isometric matching with the heat kernel. Computer Graphics Forum **29**(5), 1555–1564 (2010)
26. Patanè, G., Falcidieno, B.: Multi-scale feature spaces for shape processing and analysis. In: SMI, pp. 113 –123 (2010)
27. Reuter, M., Wolter, F.E., Peinecke, N.: Laplace-beltrami spectra as 'shape-dna' of surfaces and solids. Comput. Aided Des. **38**(4), 342–366 (2006)
28. Seo, S., Chung, M., Vorperian, H.: Heat kernel smoothing using laplace-beltrami eigenfunctions. In: MICCAI, pp. 505–512 (2010)
29. Sheffer, A., Kraevoy, V.: Pyramid coordinates for morphing and deformation. 3DPVT pp. 68–75 (2004)
30. Su, Z., Wang, H., Cao, J.: Mesh denoising based on differential coordinates. In: SMI, pp. 1–6 (2009)
31. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. In: Symposium on Geometry Processing, pp. 1383–1392 (2009)
32. Tierny, J., Vandeborre, J.P., Daoudi, M.: Topology driven 3d mesh hierarchical segmentation. In: SMI, pp. 215–220 (2007)
33. Vaxman, A., Ben-Chen, M., Gotsman, C.: A multiresolution approach to heat kernels on discrete surfaces. ACM Trans. Graph. **29**(4), 1–10 (2010)
34. Zaharescu, A., Boyer, E., Varanasi, K., Horaud, R.: Surface feature detection and description with applications to mesh matching. In: CVPR, pp. 373–380 (2009)
35. Zou, G., Hua, J., Lai, Z., Gu, X., Dong, M.: Intrinsic geometric scale space by shape diffusion. TVCG **15**(6), 1193–1200 (2009)

Appendix A: The proof of parallel property

Given any vertex v_i, its normal $n_i = (n_{ix}, n_{iy}, n_{iz})$, its neighboring vertices $\{v_j\}$, and the normalized normal weights $\{\omega_{ij}\}$, we will prove that the NCC δ_i is parallel with the normal n_i. That is, $\delta_i = \eta n_i$, where η_i is a constant.

First, there exists a plane P' with normal n_i passing through the point $\tilde{v_i} = \sum_{j \in N(i)} \omega_{ij} v_j$, whose points on such plane satisfy

$$P' = n_{ix}x + n_{iy}y + n_{iz}z - n_i \cdot \sum_{j \in N(i)} \omega_{ij} v_j = 0. \tag{27}$$

Second, let us project v_i and $\{v_j\}$ onto the plane P', denoted as $v_i{}'$ and $\{v_j{}'\}$, respectively. According to the reproduction property of mean value coordinates, we have

$$v_i' = \sum_{j \in N(i)} \omega_{ij} v_j', \qquad v_j' = v_j + \lambda_j n_i, \tag{28}$$

where λ_j is a constant. Obviously,

$$v_i - v_i' = \lambda_i n_i, \tag{29}$$

where λ_i is also a constant. Plug Eq. (28) into Eq. (29), we have

$$v_i - \sum_{j \in N(i)} \omega_{ij} v_j = \lambda_i n_i + n_i \cdot \sum_{j \in N(i)} \omega_{ij} \lambda_j. \tag{30}$$

It is easy to obtain the following equation from Eq. (27)

$$\sum_{j \in N(i)} \omega_{ij} \lambda_j = 0.$$

Finally, we have

$$v_i - \sum_{j \in N(i)} \omega_{ij} v_j = \lambda_i n_i, \tag{31}$$

where the left-hand side of Eq. (31) is the NCC δ_i, and $\eta_i = \lambda_i$ is the magnitude of δ_i. Hence, the NCC are parallel with the corresponding vertex normals. □