Anisotropic Elliptic PDEs for Feature Classification

Shengfa Wang, Tingbo Hou, Shuai Li, Zhixun Su, and Hong Qin, *Senior Member*, *IEEE*

Abstract—The extraction and classification of multitype (point, curve, patch) features on manifolds are extremely challenging, due to the lack of rigorous definition for diverse feature forms. This paper seeks a novel solution of multitype features in a mathematically rigorous way and proposes an efficient method for feature classification on manifolds. We tackle this challenge by exploring a quasi-harmonic field (QHF) generated by elliptic PDEs, which is the stable state of heat diffusion governed by anisotropic diffusion tensor. Diffusion tensor locally encodes shape geometry and controls velocity and direction of the diffusion process. The global QHF weaves points into smooth regions separated by ridges and has superior performance in combating noise/holes. Our method's originality is highlighted by the integration of locally defined diffusion tensor and globally defined elliptic PDEs in an anisotropic manner. At the computational front, the heat diffusion PDE becomes a linear system with Dirichlet condition at heat sources (called seeds). Our new algorithms afford automatic seed selection, enhanced by a fast update procedure in a high-dimensional space. By employing diffusion probability, our method can handle both manufactured parts and organic objects. Various experiments demonstrate the flexibility and high performance of our method.

Index Terms—Diffusion tensor, elliptic PDE, quasi-harmonic field, feature classification

1 Introduction

Feature extraction and classification have been of great practical importance in many graphics tasks and applications, with ever-increasing interest in recent years. Extensive studies on feature extraction, while continuing for more than a decade, have been gaining momentum, because features can assist recognition, deformation, parameterization, segmentation, shape analysis and understanding, and many more [1], [2], [3]. Features can be classified into multiple types that may include point feature, curve feature, patch feature, and so on. From the unified viewpoint of feature extraction and model segmentation, the central task of feature classification is to weave points into homologous features of different categories, whose union collectively comprise the original shape. To aid downstream graphics applications, the extraction and classification of multitype features must be of relevance to intrinsic geometry structure. The fundamental goal of this paper is to advocate an integrated strategy for feature identification, extraction, and clustering, and develop a robust and efficient method to classify multitype features of curved geometry.

Local geometric attributes, such as curvature, normal, and other surface measurements [4], [5], are frequently used to detect features. They are simple and intuitive, but suffer from noisy perturbation and incomplete information. As a local geometry description, tensor voting (TV) theory has demonstrated great advantages in modeling tasks such as feature detection, clustering, and recognition [6], [7], [8], [9]. The voting tensor is a local attribute with rich information of local geometry, which can explicitly determine the latent classification of a vertex by analyzing its eigenvectors. It has advantages over commonly used local geometric attributes. Nevertheless, certain limitations still exist. It is sensitive to noise, resulting in degraded performance for distinguishing weak features from noise. Due to the lack of knowledge for global shape information, tensor voting alone falls short of distinguishing different patch features.

Diffusion process, which is intrinsically related to the probability of random walks (RWs) [10], [11], is a powerful tool in tackling many graphics problems. It elegantly bridges the large gap between local and global behaviors via time scale and has conducted a great accomplishment in the rapid development of heat diffusion algorithms on manifolds [12], [13], [14], [15], [16]. Recent works oftentimes concentrate on dynamic solutions of the partial differential equation of heat diffusion. It typically requires global eigenfunctions of the Laplace-Beltrami operator and convolutions with heat kernels, which is very time-consuming. However, when the diffusion arrives at its steady state, time variable can be omitted in the notation. The diffusion problem then reduces to an elliptic PDE that is a linear system constrained by the heat sources. Moreover, most previous heat diffusion processes are designed isotropically, based on isotropic heat kernels on manifolds. For the problem of feature analysis, directly

- *S. Wang is with the School of Software Technology, Dalian University of Technology, Economy and Technology Development Area, Dalian City, 116620, China. E-mail: shengfawang@gmail.com.*
- *S. Li is with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, No.37 XueYuan Road, HaiDian District, Beijing, 100083, China. E-mail: lishuaiouc@126.com.*
- *Z. Su is with the School of Mathematic Science, Dalian University of Technology, NO. 2 Linggong Road, Ganjingzi Street, Dalian, 116024, China. E-mail: zxsu@dlut.edu.cn.*
- *T. Hou and H. Qin are with Department of Computer Science, Stony Brook University, Stony Brook, NY 11794.*
  E-mail: {thou, qin}@cs.stonybrook.edu.

Manuscript received 17 Oct. 2012; revised 19 Jan. 2013; accepted 22 Feb. 2013; published online 28 Feb. 2013.
Recommended for acceptance by P. Cignoni.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2012-10-0232.
Digital Object Identifier no. 10.1109/TVCG.2013.60.

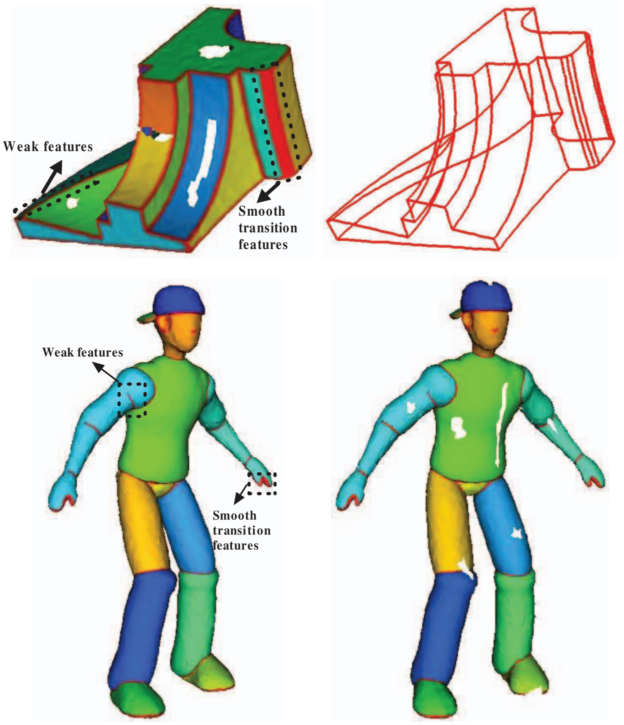


Fig. 1. Feature extraction and classification on models with noise (10 percent random noise) and holes ($c_f = 0.025$, $\delta_D = 0.16$). Different patch features are decorated in different colors, while curve and point features are colored in red. The connected curve features of Fandisk are highlighted as a wireframe representation.

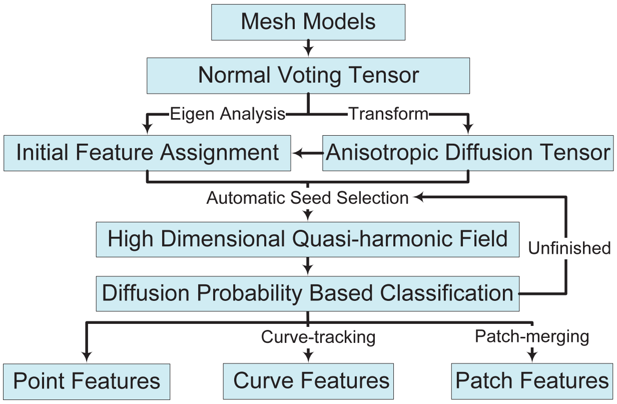


Fig. 2. The functional pipeline of our new method.

using the diffusion process will naturally give rise to the smooth transition between nearby regions, without having evident clues on feature types and how they are weaved to dictate a globally meaningful structure. To tackle this challenge, an anisotropic diffusion that has the capability of controlling diffusion directions by assigning weighted diffusion operators locally is much more favorable. In addition, the steady state of the heat diffusion corresponds to a quasi-harmonic field (QHF) generated by an elliptic PDE with the weighted diffusion operators.

In this paper, we explore a multitype feature classification based on diffusion tensor weighted quasi-harmonic field, which collectively inherit the advantages of local geometric tensors and global diffusion. The local geometric tensor is a diffusion tensor used to control a global anisotropic diffusion process on manifolds. Another advantage of this approach is that it is tunable through a few intuitive parameters and is stable under near-isometric deformations. Since these features have no rigorous definitions, we give an intuitive explanation. From the diffusion's point of view, a patch feature can be defined as a collection of piecewise smooth regions weaved together by the diffusion in a QHF. A curve feature is a curve where the QHF exhibits certain types of discontinuity. We may consider a sharp edge as a good example of curve features. A point feature is an isolated point scattered in isolation across the QHF. A transition (combination) feature can be viewed as a region that is littered on manifolds dominated by the above features. Fig. 1 highlights multitype features extracted from models with noise and holes. The weak features (nonprominent features) and smooth transition features (cluster of mixed feature vertices) can also be detected. Fig. 2 illustrates the pipeline of our approach. First, we transform the normal voting tensor to diffusion tensor and use it to initially assign feature types for vertices. A high-dimensional QHF is computed according to the seeds that are enforced as constraints in such space. Note that our seeds are automatically selected during the classification procedure, where user interaction is also supported. Then, a diffusion probability (DP)-based feature classification is exploited to classify vertices into features of different types, such as point features, curve features, and patch features. Finally, the curve-tracking and postprocessing procedures are applied. Our approach can handle 2D manifolds of arbitrary topology effectively. The salient contributions of this paper include

- We formulate a versatile diffusion tensor, which can be used to control the anisotropic diffusion, distinguish weak features from noise, and guide curve tracking, and so on.
- We devise a probability metric-based seed selection mechanism, which automatically determines the position and the number of seeds according to the shape information.
- We convert the diffusion problem to a linear system with boundary value subject to the seed constraints. Then, a high-dimensional quasi-harmonic field is obtained by fast updating the elliptic PDEs in such space.
- We develop a classification algorithm based on diffusion probability, which can be directly computed from the high-dimensional quasi-harmonic field.
- We propose a complete and robust framework for multitype feature extraction and classification that elegantly integrates the local diffusion tensor and global diffusion. It is tunable through a few intuitive parameters and is stable under near-isometric deformations.

The remainder of this paper is organized as follows: We briefly review related work in Section 2. We introduce the theory and property of the diffusion tensor in Section 3. The theoretical foundation and algorithmic steps of the proposed method are detailed in Section 4, including feature initialization, quasi-harmonic field construction, fast update, and feature classification. We demonstrate our experimental results from various aspects and discuss current limitations of our method in Section 5. Finally, we conclude our work in Section 6.

2 Background Review

Feature detection and classification can be augmented by using the knowledge from differential geometry [17], [18]. To accurately compute differential properties, both local estimation techniques [19], [20], [21] and global surface fitting [22], [23] are employed. The local estimation carries out computation in neighboring triangles of a point for fast computation. In general, the global surface fitting can enable the calculation of high-order derivatives and curvature extrema more accurately than local estimation. However, it is difficult for these methods to recognize and classify different types of features. As a viable solution, the tensor voting theory [6], [24] is adopted. Sun et al. [7] defined the normal voting tensor of a vertex on a triangular mesh by considering the unit normal vectors of neighboring triangles. This method can only recognize regions bounded by high-curvature borders. Lavoué et al. [8] introduced a method that can detect weak features and smooth transition features by using curvature tensor. Since the estimation of curvature is sensitive to noisy and sampling artifacts, it tends to overcut the entire mesh into many small pieces. Kim et al. [9] developed an alternate method by adopting the tensor voting theory, which can detect both sharp edges and transition features. However, patch features are not clustered into meaningful pieces. Besides, only local information is involved in this method, resulting in unstableness to noisy input. Wang et al. [25] solved this problem using tensor-driven diffusion for feature classification. However, the method can only handle rigid CAD models because of the random selection of seeds and the very strict feature classification algorithm. Based on the characteristics of intersection curves with blowing bubbles, Mortara et al. [26] proposed a method to locally classify vertices into a few types, which is different from our approach that takes global shape of features into account. To extract the global shape of features, ideas from mathematical morphology have been extended to surfaces [27]. However, they did not further classify the features and manipulate them corresponding to their types. Later, Lai et al. [28] proposed a feature region extraction and classification method based on a remeshing algorithm and feature sensitive metric, at the cost of topology changes. Sunkel et al. [29] learned line features from the user-specified input with training examples, but they could not find complex features with cycles. Readers can refer to the survey [30] for other feature detecting methods on meshes.

Clustering can also be interpreted as a part of feature classification, which segments a shape to several patches. From the feature's perspective, a clustering should partition a mesh into parts characterized by similarity, with boundaries as salient curves. In graphics applications, there is a large literature on clustering, such as k-means [31], [32], region growing [33], [34], [35], spectral clustering [36], mean shift [37], [38], random walks [10], [11], variational methods [39], [40], [41], and so on. Katz and Tal [31] adopted fuzzy clustering and minimal boundary cut to obtain smoother boundaries between k-mean clusters. Lai et al. [32] combined integral and statistical quantities to segment meshes with noisy or repeated patterns. One of the drawbacks of these algorithms is that they have to compute pairwise distances, which is extremely time-consuming for large meshes. Sorkine et al. [33] addressed mesh segmentation by greedy region growing, while optimizing different criteria. The easy mesh cutting algorithm [34] is introduced to cluster the similar regions using region growing, which heavily relies on seed positions and is, therefore, sensitive to noise. Spectral clustering [36] is also adopted for mesh segmentation. But the clusters are not aware of features and their spatial relationship. Yamauchi et al. [37] employed mean shift clustering of surface normals, for which computational complexity may be a prominent difficulty. Xiao and Liu [38] used K-D tree to accelerate the algorithm. However, these methods tend to overaggressively segment a mesh into more pieces than what are expected or desired. Lai et al. [10] exploited random walks to segment meshes by solving a linear system. However, since the matrix of random walks is not symmetric, it does not admit fast Cholesky factorization. Also, they did not consider weak features. Constrained random walk [11] is developed to obtain smooth cutting contours. However, the cutting only occurs between two clusters. Cohen-Steiner et al. [39] and Wang [40] casted shape approximation as a variational geometric partitioning problem. They can well identify the planar regions; however, they also overaggressively segmented the model into more pieces than what we are expecting. Yan et al. [41] exploited the quadric surface fitting to segment a mesh using the variational method. Their method worked well for tessellated CAD models, but was not versatile for general models. Zhang et al. [42] extended the Mumford-Shah model based on total variation. They automatically determined the number of segments and supported users' interactive operation, but failed at some models without clear geometry edges. Au et al. [43] located concave creases and seams using a set of concavity-sensitive scalar fields, but their method relies on local shape information and is not robust. Moreover, their method is not suitable for patch-type segmentation. We refer readers to the excellent survey papers [44], [45] for the comprehensive references and comparison on the topic of clustering.

3 Anisotropic Diffusion Tensor

In this section, we introduce an anisotropic diffusion tensor that controls both the velocity and the direction of a global diffusion. It is derived from the normal voting tensor, which is robust and can be easily extended to a higher dimensional space, but has a different formulation with more discriminative power enabled by the anisotropic diffusion. We further explore its versatile capacity to detect weak features, resist noise, and guide posttreatment procedure.

A normal voting tensor $\mathbf{NT}(v_i)$ of a vertex v_i can be computed by the sum of the weighted covariance matrices [6], [7], [9],

$$\mathbf{NT}(v_i) = \sum_{t_j \in N_t(v_i)} \mu_j \mathbf{n}_{t_j} \mathbf{n}_{t_j}^T, \quad (1)$$

where t_j is a triangle, $N_t(v_i)$ denotes the set of neighboring triangles of v_i, $\mathbf{n}_{t_j}$ is the normal of triangle t_j, and μ_j is the weight coefficient (set as 1 in our work).

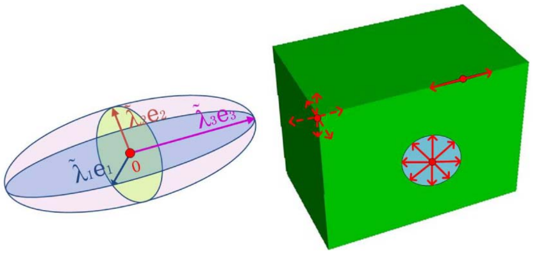

$\tilde{\lambda}_2\mathbf{e}_2$
$\tilde{\lambda}_3\mathbf{e}_3$
$\tilde{\lambda}_1\mathbf{e}_1$
0

Fig. 3. The illustration of diffusion tensor and its propagation. Left: The diffusion velocities at one vertex along all the directions can be illustrated by an ellipsoid. Right: The principal diffusion directions (denoted as red arrows) of three different vertices on a cube are shown. The dash lines initiated from corner vertex represent the directions with the diffusion velocities very close to zero.

Since the normal voting tensor is a positive semidefinite tensor with second order, it can be diagonalized by eigenvalues ($\lambda_1 > \lambda_2 > \lambda_3 \geq 0$) and reformulated by a spectral representation

$$\mathbf{NT}(v_i) = \lambda_1 \mathbf{e}_1 \mathbf{e}_1^T + \lambda_2 \mathbf{e}_2 \mathbf{e}_2^T + \lambda_3 \mathbf{e}_3 \mathbf{e}_3^T, \tag{2}$$

where $\mathbf{e}_i$ is the corresponding eigenvector of λ_i, $i = 1, 2, 3$.

The three eigenvectors of a normal voting tensor are orthogonal, and the eigenvalues characterize the diffusion velocities along the directions of the corresponding eigenvectors. Directly adopting the normal voting tensor as the diffusion tensor will lead to fast diffusion when crossing sharp edges, and slow diffusion when traveling along them. This is, however, exactly opposite to our goal. Therefore, we design a new diffusion tensor to assist the task of classifying vertices subject to different types, given by

$$\mathbf{D}(v_i) = \widetilde{\lambda}_1 \mathbf{e}_1 \mathbf{e}_1^T + \widetilde{\lambda}_2 \mathbf{e}_2 \mathbf{e}_2^T + \widetilde{\lambda}_3 \mathbf{e}_3 \mathbf{e}_3^T, \tag{3}$$

where

$$\widetilde{\lambda}_i = \exp\left(-\frac{\lambda_i}{\delta_D}\right), i = 1, 2, 3, \tag{4}$$

with diffusion parameter δ_D that controls diffusion velocities. The smaller the δ_D, the harder the heat diffuses through the hindrance, such as sharp edge, and vice versa. Intuitively speaking, we construct an ellipsoid at each vertex that encodes the direction and velocity of diffusion, as illustrated in Fig. 3 (left). According to the theory of Rayleigh quotient [46], the diffusion velocity from the vertex v_i along a vector $\mathbf{e}$ can be expressed as

$$vel(v_i, \mathbf{e}) = \frac{\mathbf{e}^T \mathbf{D}(v_i) \mathbf{e}}{\mathbf{e}^T \mathbf{e}}. \tag{5}$$

It can be interpreted as the length of the vector projection onto the ellipsoid.

Usually, the principal diffusion direction is the most informative one, which is defined as the direction corresponding to the maximal diffusion velocity. For a vertex on a plane, all the directions embedded on the plane are its principal diffusion directions, since all the diffusion velocities are equal. For a vertex on a sharp edge, the direction along the edge is its principal diffusion direction. For a vertex on a corner, all the directions are considered to be its principal diffusion directions. However, the velocities along all the directions are extremely small. Therefore, almost no heat can flow in or out. Fig. 3 (right) shows the principal diffusion directions and the propagation modes of the three types of vertices. The principal diffusion directions are also used to distinguish weak features from noise and guide the feature curve growing and merging process.

4 Our Novel Method

Given a triangular mesh, our method classifies all the vertices into features with different types. Our new method is founded upon the computation of anisotropic elliptic PDEs subject to Dirichlet boundary conditions. It comprises four steps: initial feature analysis, numerical construction of the anisotropic elliptic PDEs, feature classification, and curve tracking and postprocessing.

4.1 Feature Initialization

An initial type is assigned to each vertex according to the principal diffusion directions and the eigenvalues of its structure tensor. This initial assignment conducts a general classification of vertices and extracts vertices with salient characteristics, such as corners and sharp edges. The rest will be further classified in subsequent steps.

According to the eigen-analysis and the neighboring relationship, we assign vertices to different initial types: planar vertices, corner vertices, sharp vertices, weak feature vertices, and noisy vertices. Compared with the method in [9], our method has the ability to distinguish weak vertices from noise. Since the relative difference between eigenvalues is crucial for classifying vertices, the eigenvalues are normalized to bring consistency into different data using

$$\frac{\lambda_i}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}, \; i = 1, 2, 3. \tag{6}$$

For convenience, all the eigenvalues mentioned later are normalized, and we still denote them as λ_i.

The types of planar vertices, corner vertices, and sharp vertices can be directly determined by the corresponding eigenvalues due to the tremendous differences among them. To discriminate weak vertices and noisy vertices, we design a criterion, named *neighboring vertex coincidence* (NVC), by considering more neighbors. It utilizes the fact that a feature vertex often has neighboring vertices with similar characteristics, while a noisy vertex and its neighboring vertices usually have different principal diffusion directions. Given a vertex v_i not belonging to planar vertices, we say the vertex conforms to the NVC criterion, if it satisfies either of the following two conditions: 1) We put v_i into a front propagation set. Along its principal diffusion direction, there are nonplanar vertices having similar principal diffusion directions (with intersecting angles less than 15 degree) in its neighbors. If such coincident vertex exists and is not a corner vertex, we mark it as the new front and keep this front tracking procedure going. The number of found coincident vertices is greater than 2. 2) There are two nonplanar neighbor vertices having similar principal diffusion directions, which are also coincident with the corresponding edge that connects v_i and the neighbor vertex. Fig. 4 shows an example of using the NVC criterion to distinguish the weak features from noise. The algorithm of initial feature assignment is documented in Algorithm 1.

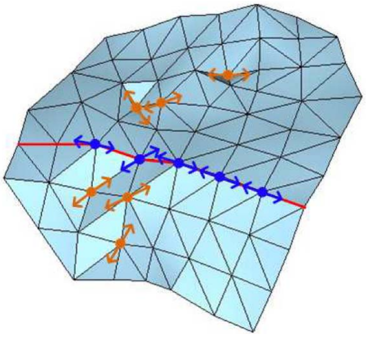


Fig. 4. Weak features (blue dots) and noisy vertices (orange dots) can be clearly separated via the NVC criterion. The arrows are the corresponding principal diffusion directions and the red line is the weak feature line.

The parameter c_f controls the boundary interface separating planar vertices and other ones, which has a large influence on curve features, and we call it the curve feature parameter and will further discuss it in the result section. The parameter c_c and c_s controls the selection of corner vertices and sharp vertices, respectively, and they have little influence on the classification results and are determined empirically. However, a proper parameter setup can increase efficiency, this is because once the sharp vertices are determined, they no longer need further processing. Moreover, since the eigenvalues are normalized, the parameters in the algorithm are consistent for different models.

```
Algorithm 1: Feature Initialization.
Parameters: $c_f = 0.03$, $c_c = 0.1$, $c_s = 0.3$.
for each vertex do
    if $\lambda_3 > c_c$ then
        mark corner vertex;
    else if $\lambda_2 < c_f$ then
        mark planar vertex;
    else if $\lambda_2 > c_s$ & $\lambda_3 < c_c$ & NVC then
        mark sharp vertex;
    else if $c_f \le \lambda_2 \le c_s$ & $\lambda_3 < c_c$ & NVC then
        mark weak vertex;
    else
        mark noisy vertex;
    end
end
```

Since we do not categorize noise as a feature type, a separate process is taken to reassign feature types for weak vertices and noisy vertices. Considering that the diffusion is a global PDE that has a built-in resistance to noise, a simple smoothing and enhancement procedure suffices for this purpose. We process a noisy vertex by smoothing the second eigenvalue of the voting tensor according to its neighbors, given by

$$\lambda_2(v_i) = \frac{\sum_{j\in N(i)} vel(v_j, v_i - v_j)\lambda_2(v_j)}{\sum_{j\in \widetilde{N}(i)} vel(v_j, v_i - v_j)}, \tag{7}$$

where $\lambda_2(v_i)$ is the second eigenvalue of the voting tensor of v_i, $vel(.,.)$ is the diffusion velocity defined in (5), and $N(i)$ is the set of neighboring vertices of v_i. Then, we enhance the weak vertices by simply enlarging the second eigenvalue of the voting tensor by a factor of 10. If a weak vertex satisfies only the second condition of NVC, we need to further replace its diffusion tensor by the average diffusion tensors of the coincident neighbors. After this procedure, noisy vertices are further classified into weak vertices or planar vertices, while weak vertices are enhanced and reclassified into the sharp vertices. After the initial feature assignment, we mark corner vertices as point features, and the sharp vertices as curve features. From the global point of view, the boundary of a patch feature can also be treated as global curve features. The planar vertices will be further classified into different patch features and curve features.

4.2 Quasi-Harmonic Fields

In our method, we utilize the QHF to provide the stable distribution of heat with some heat sources (seeds), which are Dirichlet conditions for elliptic PDEs. The heat diffusion over a manifold M is governed by the heat equation. We formulate the weighted diffusion process as

$$\begin{cases} \dfrac{\partial u(v,t)}{\partial t} = -div(\widetilde{\mathbf{D}}\nabla u(v,t)) & t \in R^+ \\ u(v,t) = c(v) & v \in S \\ u(v,0) = 0 & v \in \text{others}, \end{cases} \tag{8}$$

where the diffusivity $\widetilde{\mathbf{D}}$ is a 3×3 symmetric matrix, S is a set of seeds, and $c(v)$ is the fixed value of seed v. The weight matrix $\widetilde{\mathbf{D}}$ serves for two purposes: encoding diffusion tensor $\mathbf{D}$, and characterizing geometric difference between neighboring vertices. These local attributes are crucial for our feature classification, which will be addressed next.

From a global perspective, we allow heat diffusion to reach its equilibrium and consider the stable state of the weighted diffusion in (8). When the diffusion has reached its stable state, time t is omitted in the notation. Then, (8) reduces to an elliptic PDE

$$\begin{cases} div(\widetilde{\mathbf{D}}\nabla u(v)) = 0 \\ u(v) = c(v) & v \in S, \end{cases} \tag{9}$$

whose solution is a QHF. The discrete formulation of (9) can be written into matrix form:

$$\mathbf{LF} = 0, \tag{10}$$

subject to Dirichlet conditions $\mathbf{F}(v) = c(v), v \in S$, where $\mathbf{L}$ is a $n \times n$ coefficient matrix, and $\mathbf{F}$ is a quasi-harmonic field. The coefficient matrix $\mathbf{L}$ has elements

$$\begin{cases} \mathbf{L}_{ij} = -K(v_i, v_j), \\ \mathbf{L}_{ii} = \sum\limits_{j\in N(i)} K(v_i, v_j), \end{cases} \tag{11}$$

with

$$K(v_i, v_j) = \exp\left(-\frac{(v_i - v_j)^T \mathbf{D}(v_i, v_j)^{-1}(v_i - v_j)}{\delta_K}\right), \tag{12}$$

where $\mathbf{D}(v_i, v_j) = w_{ij}(\mathbf{D}(v_i) + \mathbf{D}(v_j))$, and δ_K is a penalty factor set to be the inverse of the maximal eigenvalue of diffusion tensors $\mathbf{D}(v_i)$ and $\mathbf{D}(v_j)$. The geometry-aware weight w_{ij} is used as a diffusion tensor aide, and it is defined as

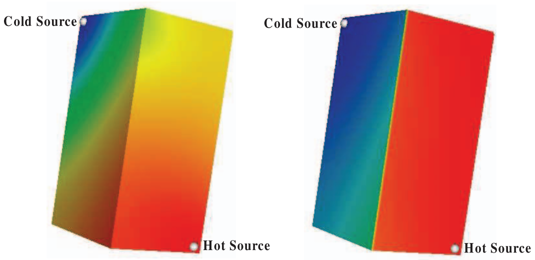


Fig. 5. The QHFs generated using common cotangent Laplace operator (left) and our anisotropic diffusion operator (right). The seeds are cold source with the constant temperature of 0 and hot source with the constant temperature of 1, respectively.

$$w_{ij} = \exp\left(-\frac{\|\mathbf{NCC}_i - \mathbf{NCC}_j\|}{\delta_G}\right), \tag{13}$$

where $\mathbf{NCC}_i$ denotes normal-controlled coordinates (NCC) [47] of vertex v_i, and δ_G is a geometry-dilation parameter that controls the influence of geometry descriptor NCC. Here, δ_G is set to be $\max\{avg_f(|\ NCC\ |), 0.3\}$, where $avg_f(|\ NCC\ |)$ is the average value of the NCCs of nonplanar vertices. We use NCC here, because they encode local geometric details of a vertex along its normal direction, and have proper behaviors for open surfaces without tangential tension. In this way, geometric differences of neighboring vertices are well categorized. Since the diffusion tensor $\mathbf{D}(\cdot)$ is positive definite and $w_{ij} = w_{ji}$ is a positive constant, the value of (12) is within interval (0,1]. Moreover, it is easy to have $K(v_i, v_j) = K(v_j, v_i)$, and we know that the coefficient matrix $\mathbf{L}$ is symmetric. Fig. 5 shows an example of our QHF with two seeds. Our QHF has shown a clear temperature discrepancy between different parts separated by the sharp edge.

4.3 Fast Update QHF in High-Dimensional Space

A QHF is the solution of an elliptic PDE (10) with Dirichlet conditions at seeds. A new seed will be automatically selected according to the current QHF if the classification is unfinished. For large meshes, it is time-consuming to solve the QHF every time after the seeds are updated. We adopt the popular *Penalty method* [48] to fast update the QHF with Dirichlet condition assigned by the seeds. Specifically, $\mathbf{L}$ is symmetric that admits fast Cholesky factorization and fast updating of Cholesky [49]. As a result, adding/removing seed constraints can be written as matrix additions. On the other hand, to adjust the quasi-harmonic field suitable for clustering, we construct a high-dimensional quasi-harmonic field in R^d using selected seeds, where d is the number of seeds. Hence, we need to put seeds in R^d to construct high-dimensional constraints, which is a $n \times d$ matrix. Each column of the constraint matrix is associated with one seed initially valued as 1, and the other seeds are set to be 0. Now, $\mathbf{F}$ becomes a $n \times d$ unknown matrix, which represents d-dimensional quasi-harmonic fields. The region where the values are most similar to that of a seed in the quasi-harmonic field is treated as a patch feature. Then, (10) can be rewritten as

$$(\mathbf{L} + \bar{\mathbf{P}})\mathbf{F} = \bar{\mathbf{P}}\mathbf{B}, \bar{\mathbf{P}} = \mathbf{P} + \mathbf{U}\mathbf{U}^T - \mathbf{R}\mathbf{R}^T, \tag{14}$$

where the $n \times n$ penalty matrix $\mathbf{P}$, the $n \times n$ modification matrices $\mathbf{U}$ and $\mathbf{R}$, and the $n \times d$ constraint matrix $\mathbf{B}$ have the following entries:

$$\mathbf{P}_{ij} = \begin{cases} \alpha & i = j \in C \\ 0 & \text{otherwise} \end{cases} \qquad \mathbf{U}_{ij} = \begin{cases} \sqrt{\alpha} & i = j \in C_i \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{R}_{ij} = \begin{cases} \sqrt{\alpha} & i = j \in C_d \\ 0 & \text{otherwise} \end{cases} \qquad \mathbf{B}_{ij} = \begin{cases} 1 & i \text{ is the } j\text{th } in\ \bar{\mathrm{C}} \\ 0 & \text{otherwise,} \end{cases}$$

with α being the penalty factor, C the indices for the previous seeds, C_i the indices for newly inserted seeds, C_d the indices of seeds to be deleted, and $\bar{C}$ the indices for the updated seeds. As for the penalty factor, we choose $\alpha = 10^8$ for all the examples in our current implementation. It may be noted that the penalty method only handles soft constraints, so the α value must be large enough to confine the values at seeds within a proper range [48].

Our seed selection is different from the previous methods. We design a probability distance-based scheme to automatically select seeds in the classification procedure. Given a quasi-harmonic field, we classify a subset of planar vertices into patch features. Then, a new seed is selected from the nonclassified planar vertices. We can obtain the updated quasi-harmonic field corresponding to the newly updated seeds. The process is repeated until all the planar vertices are classified. We compute the initial quasi-harmonic field by randomly selecting two seeds from the planar vertices with the euclidean distance larger than the radius of the model's bounding sphere. This can avoid overclassification. Meanwhile, our system also supports user interaction for seed selection, which can fulfill specific needs of feature segmentation. The seed selection and feature classification will be detailed in the following section.

4.4 Feature Classification Based on Diffusion Probability

In this part, we first give a definition, called diffusion probability. After the d-dimensional QHF F is computed from (14), the DP of a vertex v_i to the jth seed $DP(v_i, j)$ can be intuitively defined as the jth element of F_i, where $F_i = (F_{i1}, \ldots, F_{id})^T$ is the vector value of the QHF at vertex v_i. Then, we classify features by a self-adaptive algorithm based on the newly introduced concept of diffusion probability. According to our prior discussion on high-dimensional QHF and their valid value ranges, we observe that each scalar component of this d-dimension vector at any vertex is a nonnegative value between 0 and 1, it is both natural and mathematically intuitive to consider the d-dimensional vector field at a vertex as a probability distribution function (PDF) with respect to d selected seeds: The probabilities from the d corresponding seeds randomly walking (diffusing) to the current vertex. Moreover, our newly devised DP can be immediately obtained from the existing QHF without extra computational effort, at the same time it exhibits the robustness of QHF. Therefore, DP can be considered as a powerful measurement to classify vertices from the perspective of diffusion.

In the initial feature assignment, most types of vertices have already been classified except planar vertices, we only need to handle planar vertices in this step, and classify them into different patch features and curve features via our classification algorithm (Algorithm 2). The key idea is to cluster planar vertices to the corresponding seeds that have

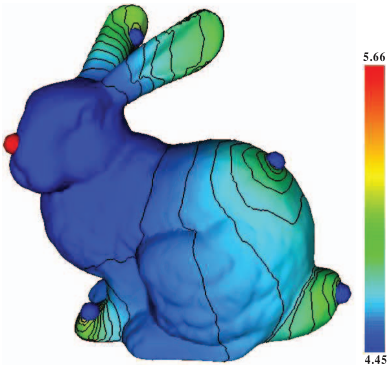


Fig. 6. An example illustrates the automatic seed selection mechanism. The color scheme denotes the value in the distance field: the sum of the probability distances from a vertex to all the existing seeds (denoted as blue balls). The isolines are constructed by connecting vertices that share the distance value. The new seed (denoted as a red ball) is highlighted at a location that has a minimum value in the distance field.

the maximum DP. Naturally, the boundary of each patch feature constructs a curve feature that reflects geometric saliency and segments the patch. Therefore, we treat boundaries of the patches as curve features in our classification algorithm. Intuitively speaking, patch features are weaved together by curve features.

```
Algorithm 2: Diffusion-probability-based Feature Classification.
for $i$=1:$n$ do
    find $[f_{ij}, j]$=$\max_j(DP(v_i, j))$;
    if $v_i$ is a non-planar vertex or $f_{ij} < 1/j + 0.1$ then
        continue;
    end
    add the vertex $v_i$ to the $j$-th patch feature, and
    assign it the color value $j$;
    remove it from planar vertices $V_P$;
end
if $V_P$ is empty then
    mark the common boundary vertices of patch
    features as curve features;
end
```

Now, we shall introduce our seed selection mechanism. To automatically select seeds, a proper metric is inevitable. Many commonly used metrics, such as geodesic distance and diffusion distance, suffice for this purpose. Nevertheless, accurately calculating these metrics on manifolds is time-consuming in principle. Since the QHF is already computed in the earlier stage, it makes a perfect sense to devise a distance metric that makes full use of it. In the d-dimensional QHF F, we define a probability distance between two vertices as

$$DP_F(v_i, v_j) = \|F_i - F_j\|_2. \tag{15}$$

For a generic family of probability distribution functions, it is easy to verify that when $d > 2$, the distance $DP_F(v_i, v_j)$ is a metric by using Minkowski's inequality. This distance metric will be further employed to merge small patch features in later stages.

We iteratively determine seeds based on this newly proposed distance measurement. During the iteration, a new seed is chosen with the minimum distance value to all the existing seeds. Fig. 6 illustrates an example of how a new seed can be automatically placed on the model and where to place such seed in an automatic way. The feature classification based on automatic seed selection is documented in Algorithm 3. Given the Cholesky factorization of coefficient matrix and the nonclassified planar vertices, the classification is achieved after a fewer iterations. Since the QHF has been computed in an earlier stage, the distance measures and their summation can be obtained directly. If the user needs to further segment large patch features into small ones, our system also supports user interaction, which can be handled in a similar way. Note that the number of selected seeds equals to the number of patch features.

```
Algorithm 3: Feature classification with automatic seed selection.
input : Cholesky factorization of the coefficient
        matrix $\mathbf{L}$, planar vertices $V_P$.
output: Classification result.
Initialize: $C = \emptyset$, $\bar{C} = \emptyset$, $C_d = \emptyset$.
while $V_P$ is not empty do
    1: set $C_i = \emptyset$, select a new seed from $V_P$ that has
    the minimum sum of distances from the existing
    seeds, and insert it to $C_i$ and $\bar{C}$.
    2: fast update the Cholesky via Eq. (14), and
    obtain the updated $\mathbf{F}$.
    3: classify the planar vertices using Algorithm 2,
    and set $C$=$\bar{C}$.
end
```

4.5 Curve Tracking and Postprocessing

Now, the vertices are classified into different features. For models with sharp edge features, we build a concise representation for a curve feature by ordering the vertices and connecting them. This representation is useful in downstream graphics applications. In previous methods [23], [50], the connection algorithms are utilized, even though they may generate many branches and intricate lines. In our work, this can be easily handled using the principal diffusion directions. Smooth feature curves are found by curve tracking along the principal diffusion directions. For an edge of M, if both endpoints belong to point features or curve features, and at least one of the principal diffusion directions is close to the edge (whose intersecting angle is less than 15 degree), the two endpoints are connected to form a line segment contributing to a curve feature. For more general models, different line features and point features may be interleaved together to form a more complicated transition region. In this case, these features are just illustrated by color rather than clustering into a connected curve.

Finally, a postprocessing step might still be needed to process small patch features. A small patch feature is recognized as the one with the number of vertices fewer than a given threshold (we set it to be 0.1 percent of the total number of vertices in this paper). It is possible that a large feature is separated by small ones due to noise or nonsalient geometry. We tackle this problem by merging a small patch into the most relevant neighboring patch (if such patch exists), whose seed has the minimum distance to the small patch. The distance of a vertex v to a patch P is defined as

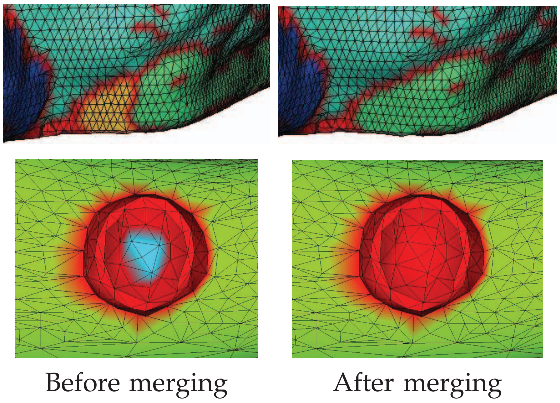


Fig. 7. Merging small patch features. The top row: A small patch is merged into a patch feature. The bottom row: A small patch is merged into a transition feature.

$$DP_F(v, P) = \min_{u \in P}\{DP_F(v, u)\}. \tag{16}$$

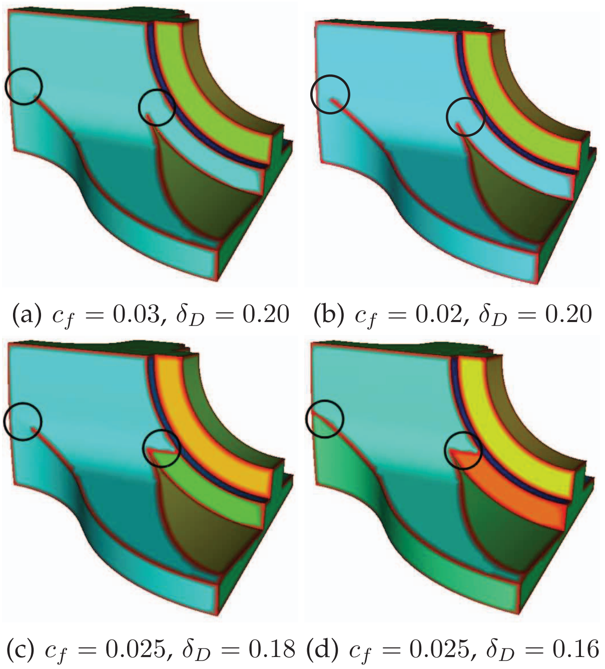
Sometimes, a small patch is surrounded by transition features. In this case, we just merge it into a transition feature. Fig. 7 shows the example of merging small patch features. Note that this postprocessing stage is rather optional and can be skipped by users, since small patches may be useful in certain applications.

5 Experimental Results and Discussions

In this section, we demonstrate the performance of our method by conducting experiments in various aspects, including multilevel classification, noisy meshes, comparison with other methods, general models, and deformable meshes. All the experiments documented in this paper are conducted on a computer with 1.6-GHz Intel Core (TM, four cores/eight threads) i7 CPU with 4-Gb RAM, where both synthetic and scanned meshes are utilized. Most computation expenses of our approach can be done in the preprocessing stage, such as the diffusion tensor computation, feature initialization, and Cholesky decomposition of the coefficient matrix. Since only the updating procedure instead of recomputation is needed in each iteration, our method is extremely efficient and can reach real-time performance. Table 1 summarizes the statistics and timing performance of fast updating of the QHF for a collection of data set used in our experiments, where $\#C_i$ and $\#C_d$ are the numbers of newly inserted seeds and deleted seeds, respectively

TABLE 1
Time Performance of Updating Quasi-Harmonic Fields with High-Dimensional Seeds

Models	# V	# C_i	# C_d	Time (sec)
Tre-twist	12800	10	10	0.001
Pawn	68000	10	10	0.009
Pawn	68000	1	1	0.001
hand	170000	10	1	0.028
hand	170000	1	0	0.002



(a) $c_f = 0.03$, $\delta_D = 0.20$ (b) $c_f = 0.02$, $\delta_D = 0.20$

(c) $c_f = 0.025$, $\delta_D = 0.18$ (d) $c_f = 0.025$, $\delta_D = 0.16$

Fig. 8. Multilevel feature classification on the Fandisk, obtained by adjusting the curve feature parameter c_f and the diffusion parameter δ_D.

Parameters and multilevel classification. Parameter selection is a challenge for versatile intelligent methods. Usually, fewer parameters and flexible multifunctional capabilities are conflicting with each other, people have to find a good tradeoff in practice. There are several parameters that need to set in our approach; however, most of them are either self-adaptive (δ_K is set to be the inverse of the maximal eigenvalue of the corresponding diffusion tensor, and $\delta_G = \max\{avg_f(\mid NCC \mid), 0.3\}$) or determined empirically ($c_c = 0.1$ and $c_s = 0.3$), and they are consistent throughout different models. We only need to pay special attention to two parameters: curve feature parameter c_f and the diffusion parameter δ_D. As we have discussed above, c_f influences curve features. In principle, the less prominent of the features, the smaller of curve feature parameter should be. δ_D controls the influence of diffusion tensor to the QHF, the less of the heat passing across the feature regions, the smaller of the diffusion parameter should be. That is, δ_D influences the patch features. Fig. 8 shows the effect of the parameters. They are set to be 0.03 and 0.16, respectively, by default if there is no special requirement on parameter setting.

Moreover, our method also supports hierarchical feature classification by interactively adjusting the curve feature parameter and the diffusion parameter to form different combinations. From coarse to fine, a large piece of patch feature may be further classified into some small pieces. Fig. 8 shows the multilevel results with different parameters. We want to particularly mention that the curve feature parameter is intuitive, users can interactively slide the value bar to obtain the desired curve features. In our ongoing work, we would like to improve the parameter setting and tuning by exploring the intrinsic relationship between the curve feature parameter and the

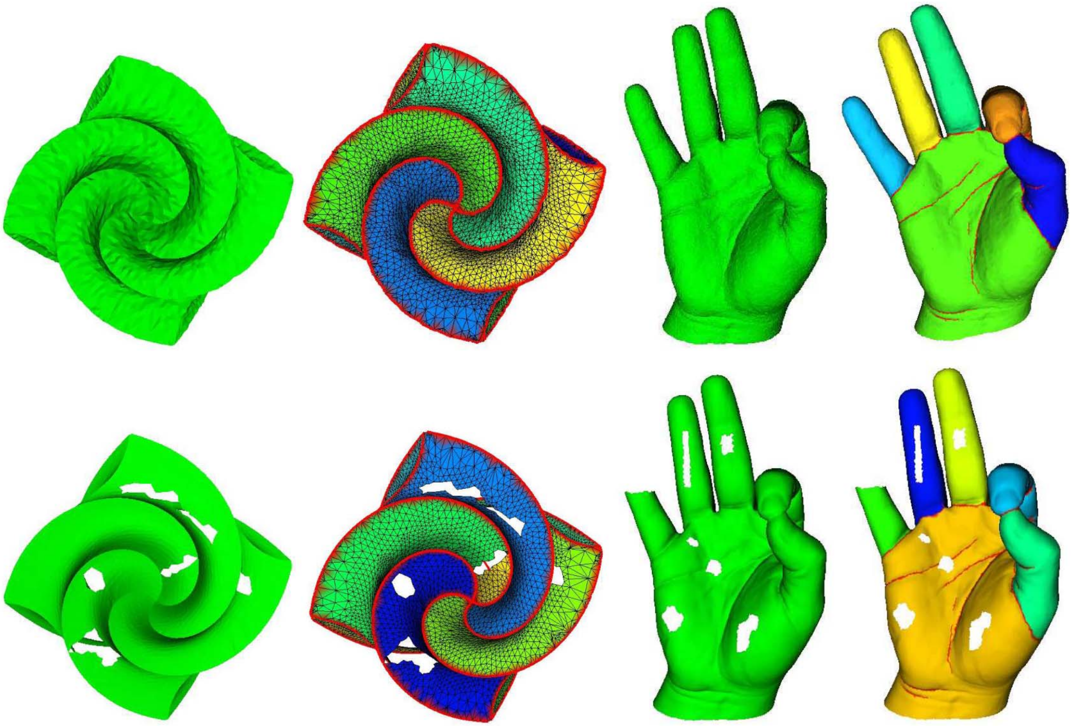


Fig. 9. Feature classification results for models with noise and holes ($c_f = 0.03$, $\delta_D = 0.16$). The first row: models (Octa-flower and Hand) with 10 percent random noise. The second row: models with holes.

diffusion parameter and devise a semiadaptive parameter selection scheme.

Classification on meshes with noise and holes. The tensor is a local attribute determined by a small neighborhood of a vertex. Therefore, only using the local tensor [9] can be easily affected by noise. To address this problem, we adopt QHFs in our method to classify features, which are stable solutions to heat diffusion. The QHFs have high resilience in combating noise and holes. Moreover, the gap between features and noise are further magnified by exploiting the

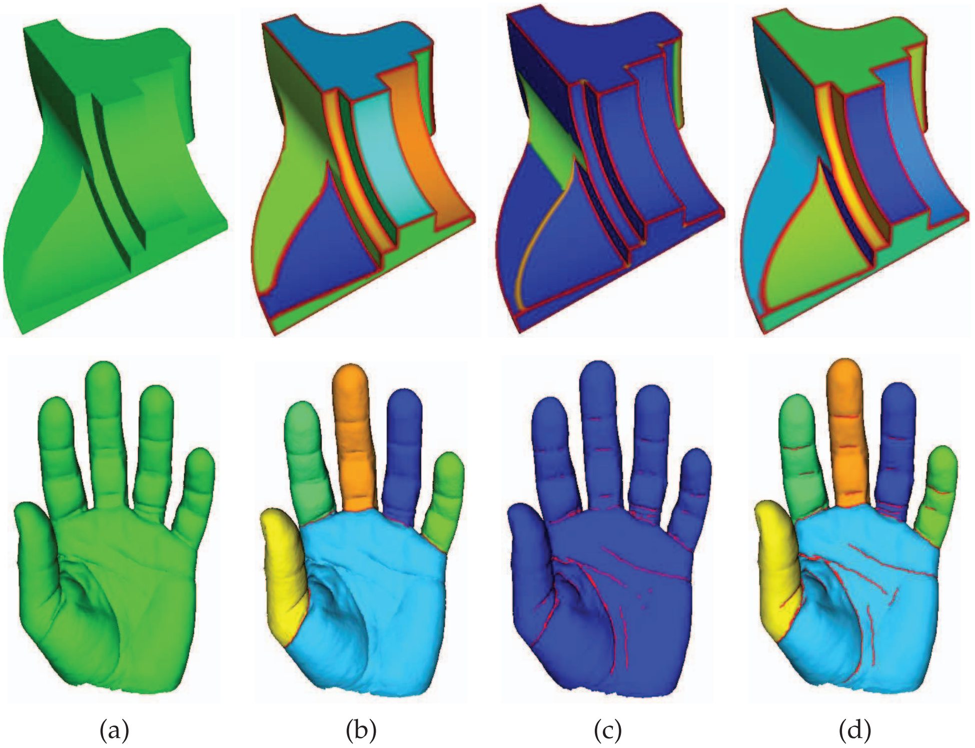


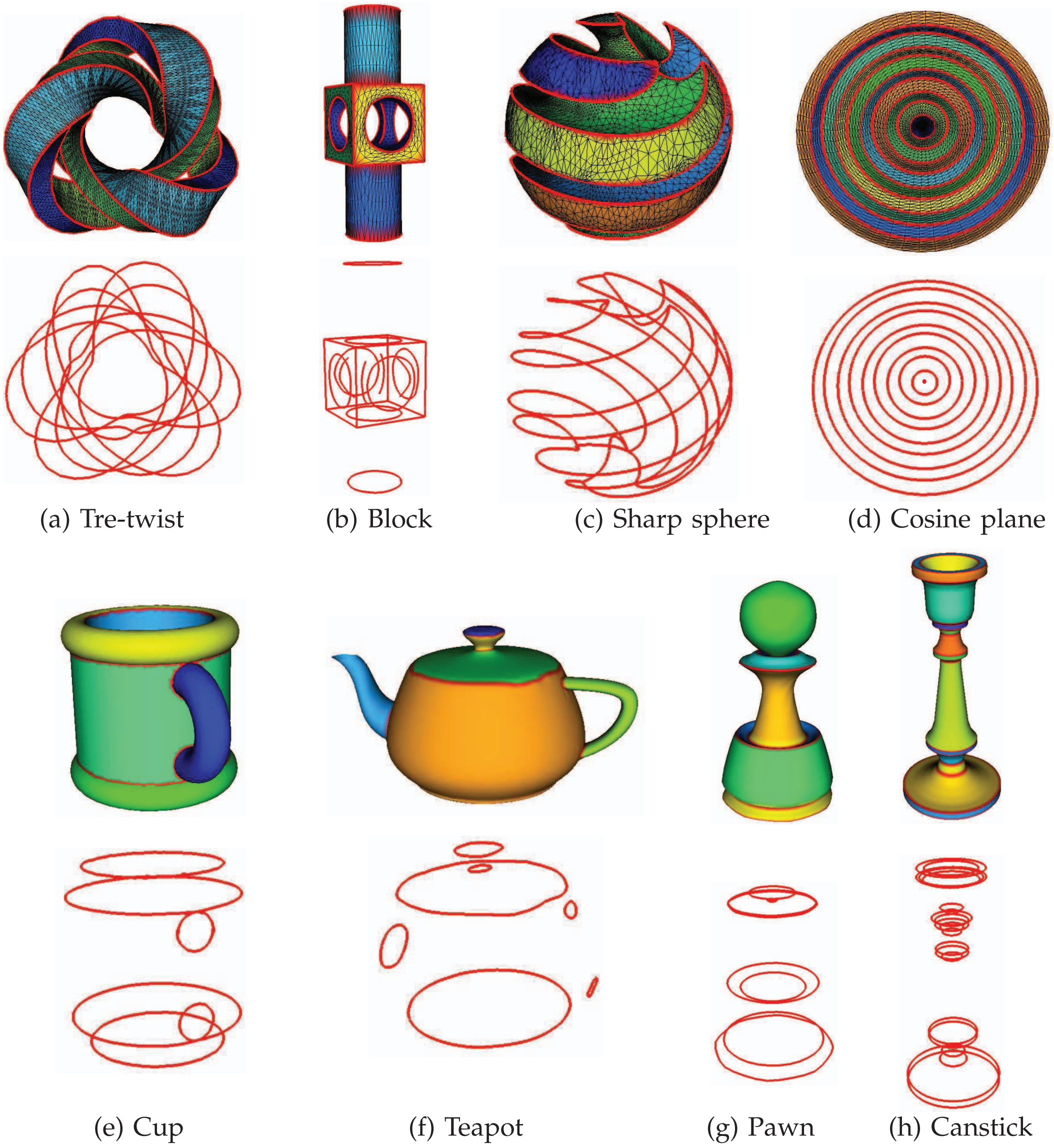
Fig. 10. Comparison of different methods. (a) Original meshes. (b) The results of random walk [10]. (c) The results of tensor voting [9]. (d) Our results ($c_f = 0.025$, $\delta_D = 0.16$).

TABLE 2
Comparison of Different Methods in Time (sec)

Models	# V	RW	TV	Ours
Fandisk	6477	0.44	3.62	0.35
Tre-twist	12800	1.78	5.31	0.98
Pawn	68000	8.64	25.23	6.24
Octopus	100000	12.05	35.82	10.22
Hand	170000	30.13	73.66	19.96

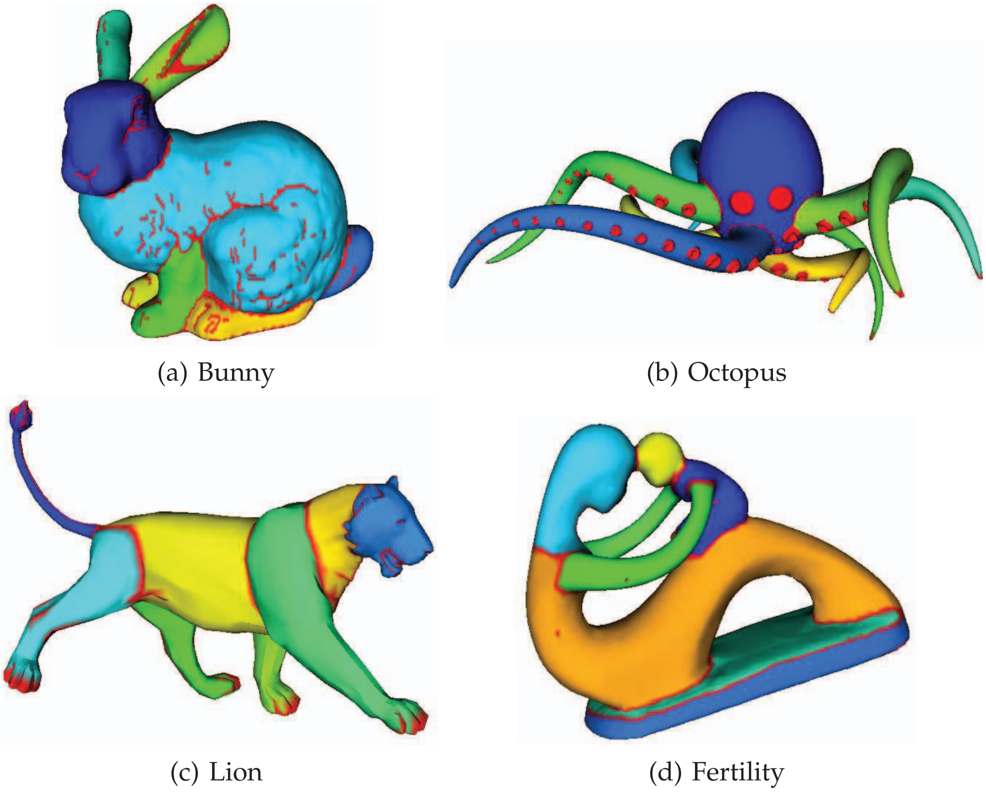
information of the diffusion tensor. Figs. 1 and 9 show the classification results under noise and holes. We add 10 percent (of average edge length) random noise to perturb vertex coordinates and punch some holes (topological noise) on the models. The small holes almost have no negative effect on the results because of the diffusion nature. When a hole is too large to allow the diffusion to pass through by traversing around the hole boundary, the corresponding patches will be further classified into smaller patches according to the hole locations (Fig. 9 (bottom left)). Note that, due to the diffusion nature of our QHF, no special processing is required for the holes and open boundaries.

Comparison and discussion. In Fig. 10, we compare our method with previous related methods: random walk [10] (b), and tensor voting [9] (c). The RW method fails to find weak features, since there is no feasible criterion to stop the random walk from passing through the weak features. Moreover, since the matrix of random walk is not symmetric, fast Cholesky factorization is not applicable, which limits its computational speed. The TV method cannot distinguish weak features from noise either. Therefore, the clustering is sensitive to noise, and much more postprocessing is unavoidable. Besides, neither could distinguish different patch features, they instead consider them collectively as one patch. Our method can classify the vertices into different features, and it can detect both the weak features and the smooth transition features for noisy


(a) Tre-twist (b) Block (c) Sharp sphere (d) Cosine plane

(e) Cup (f) Teapot (g) Pawn (h) Canstick

Fig. 11. More examples for man-made objects ($c_f = 0.03$, $\delta_D = 0.16$). The connected curve features are highlighted beneath the feature classification results.



(a) Bunny

(b) Octopus

(c) Lion

(d) Fertility

Fig. 12. Feature classification on organic models. Patch features are shown in different colors, while all other features, including point features, curve features, and transition features are shown in red ($c_f = 0.03$, $\delta_D = 0.16$).

meshes (Figs. 1 and 10d). Table 2 summarizes the running time of the algorithms used in experiments in Fig. 10. It indicates that our method performs better than the other methods in terms of both speed and classification results.

More examples including deformed models. More experimental results are shown to further demonstrate the performance of our method. Fig. 11 demonstrates some results of rigid manufactured objects. The connected curve

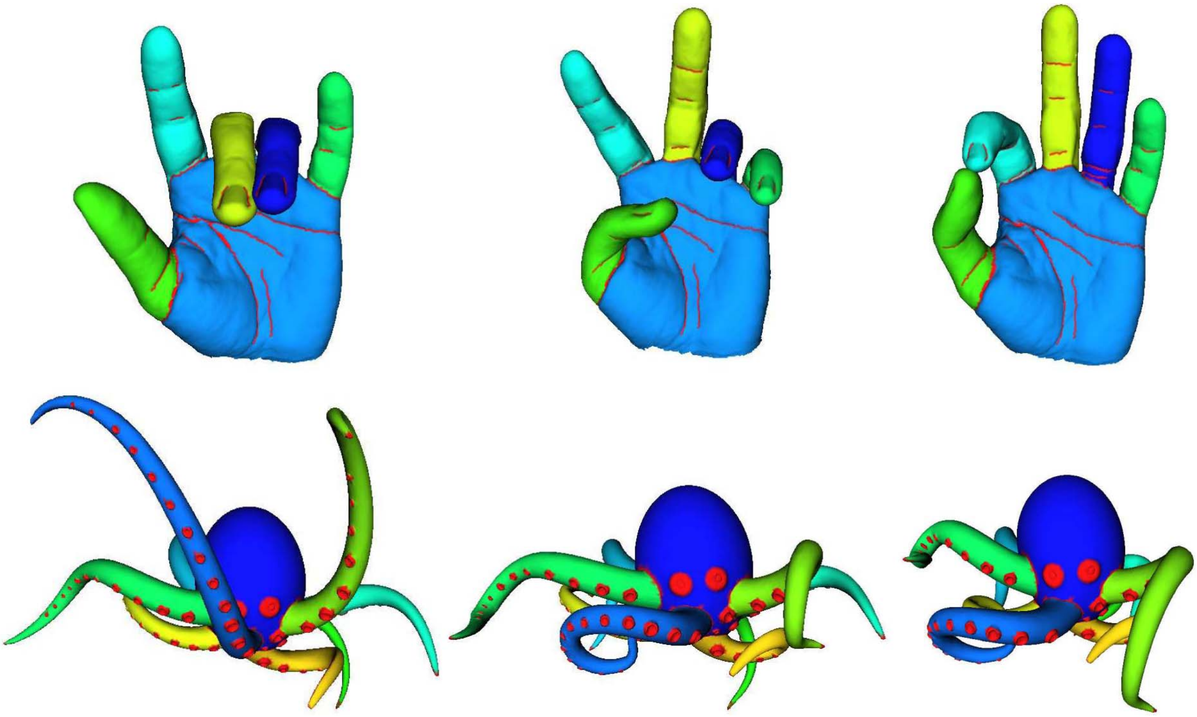


Fig. 13. Consistent classification of deformed models. The classified features are well preserved under a series of near-isometric deformations ($c_f = 0.03$, $\delta_D = 0.16$).

features are highlighted beneath the classification results. We also examine our method on natural objects, with experiments in Fig. 12. For simplicity, different patch features are shown in different colors, and all other types of features, including point features, curve features and transition features being shown in red. Near-isometric deformation usually preserves significant features of models. Thus, our method can be expected to generate consistent results under such deformation. Fig. 13 shows the classification of three deformed models. Most classification results are well preserved while undergoing deformation, except for some places where the geometric features vanish after deformation, such as some creases on the thumb.

Limitations. Since the curve features of general models are often complicated, they cannot be easily represented by wireframes (which otherwise could be easily handled for manufactured models). Also, in some extreme cases, the features in small complicated regions are hard to be discriminated. In such cases, we simply treat them as transition (or composite) features consisting of point features, curve features, and patch features, such as the eyes and suckers of octopus shown in Fig. 13. For the clarity of visualization, we just highlight them using red color in this paper. How to derive a more general representation, which considers either semantics or specific application needs, will be our future work.

6 Conclusion and Future Work

In this paper, we have articulated a new method for feature extraction and classification on meshes, based on diffusion tensor driven quasi-harmonic fields. A diffusion tensor has been locally designed to control the global anisotropic behavior of heat diffusion. Such diffusion tensor has also been utilized to eliminate noise and form feature curves. The novelty of our method centers at the elegant integration of the locally defined diffusion tensor and the globally defined quasi-harmonic field in an anisotropic manner. Moreover, we transform the nonlinear diffusion problem into a linear problem of field construction to make our method computationally efficient. A greedy feature classification process enables our method to handle both rigid manufactured parts and organic deformable objects. We believe that the fundamental ideas in this paper can propel us to pursue more real-world applications, including feature-driven shape registration, local feature-sensitive parameterization, object recognition, shape synthesis, and so on.

For immediate future work, we are planning to extend our method to handle diverse types of geometric and scientific data, such as point clouds in urban architecture modeling and volumetric data in medical imaging. Moreover, applying this approach to vector/tensor field design and feature-aware nonphotorealistic visualization deserves further investigation that could significantly broaden our method's application scope.

Acknowledgments

This research was supported in part by the Fundamental Research Fund for the Central Universities, National Natural Science Foundation of China-Guangdong Joint Fund grant U0935004, National Natural Science Foundation of China grants 60873181, 91230103, 61190120, 61190121, and 61190125, and the US National Science Foundation grants IIS-0710819, IIS-0949467, IIS-1047715, and IIS-1049448.

References

[1] Q. Wu and Y. Yu, "Feature Matching and Deformation for Texture Synthesis," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 364-367, 2004.

[2] E. Zhang, K. Mischaikow, and G. Turk, "Feature-Based Surface Parameterization and Texture Mapping," *ACM Trans. Graphics,* vol. 24, no. 1, pp. 1-27, 2005.

[3] T. Hou and H. Qin, "Efficient Computation of Scale-Space Features for Deformable Shape Correspondences," *Proc. European Conf. Computer Vision,* pp. 384-397, 2010.

[4] M. Meyer, M. Desbrun, P. Schrder, and A.H. Barr, "Discrete Differential-Geometry Operators for Triangulated 2-Manifolds," *Visualization and Mathematics,* pp. 1-26, Springer, 2002.

[5] C.C.L. Wang, "Bilateral Recovering of Sharp Edges on Feature-Insensitive Sampled Meshes," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 4, pp. 629-639, July/Aug. 2006.

[6] G. Medioni, C.-K. Tang, and M.-S. Lee, "Tensor Voting: Theory and Applications," *Reconnaissance des Formes et Intelligence Artificielle,* 2000.

[7] D.L. Page, Y. Sun, A.F. Koschan, J. Paik, and M.A. Abidi, "Normal Vector Voting: Crease Detection and Curvature Estimation on Large, Noisy Meshes," *Graphical Models,* vol. 64, nos. 3/4, pp. 199-229, 2002.

[8] G. Lavoué, F. Dupont, and A. Baskurt, "A New Cad Mesh Segmentation Method, Based on Curvature Tensor Analysis," *Computer-Aided Design,* vol. 37, no. 10, pp. 975-987, 2005.

[9] H.S. Kim, H.K. Choi, and K.H. Lee, "Feature Detection of Triangular Meshes Based on Tensor Voting Theory," *Computer-Aided Design,* vol. 41, no. 1, pp. 47-58, 2009.

[10] Y. Lai, S. Hu, R.R. Martin, and P.L. Rosin, "Rapid and Effective Segmentation of 3D Models Using Random Walks," *Computer Aided Geometric Design,* vol. 26, no. 6, pp. 665-679, 2009.

[11] J. Zhang, J. Zheng, and J. Cai, "Interactive Mesh Cutting Using Constrained Random Walks," *IEEE Trans. Visualization and Computer Graphics,* vol. 17, no. 3, pp. 357-367, Mar. 2011.

[12] M. Desbrun, M. Meyer, P. Schröder, and A.H. Barr, "Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow," *Proc. ACM SIGGRAPH,* pp. 317-324, 1999,

[13] U. Clarenz, U. Diewald, and M. Rumpf, "Anisotropic Geometric Diffusion in Surface Processing," *Proc. Visualization,* pp. 397-405, 2000.

[14] J. Sun, M. Ovsjanikov, and L. Guibas, "A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion," *Computer Graphics Forum,* vol. 37, no. 10, pp. 1383-1392, 2009.

[15] T. Hou and H. Qin, "Continuous and Discrete Mexican Hat Wavelet Transforms on Manifolds," *Graphical Models,* vol. 74, no. 4, pp. 221-232, 2012.

[16] T. Hou and H. Qin, "Admissible Diffusion Wavelets and Their Applications in Space-Frequency Processing," *IEEE Trans. Visualization and Computer Graphics,* vol. 19, no. 1, pp. 3-15, Jan. 2013.

[17] A.G. Belyaev, E.V. Anoshkina, and T.L. Kunii, "Ridges, Ravines and Singularities," *Topological Modeling for Visualization,* vol. 18, no. 11, pp. 375-383, 1997.

[18] K. Hildebrandt, K. Polthier, and M. Wardetzky, "Smooth Feature Lines on Surface Meshes," *Proc. Symp. Geometry Processing,* pp. 85-90, 2005.

[19] G. Stylianou and G. Farin, "Crest Lines for Surface Segmentation and Flattening," *IEEE Trans. Visualization and Computer Graphics,* vol. 10, no. 5, pp. 536-544, Sept./Oct. 2004.

[20] U. Clarenz, M. Rumpf, and A. Telea, "Robust Feature Detection and Local Classification for Surfaces Based on Moment Analysis," *IEEE Trans. Visualization and Computer Graphics,* vol. 10, no. 5, pp. 516-524, Sept./Oct. 2004.

[21] R. Gal and D. Cohen-or, "Salient Geometric Features for Partial Shape Matching and Similarity," *ACM Trans. Graphics,* vol. 25, no. 1, pp. 130-150, 2006.

[22] J. Kent, K.V. Mardia, J.M. West, and L.L. Jt, "Ridge Curves and Shape Analysis," *Proc. British Machine Vision Conf.,* pp. 43-52, 1996.

[23] Y. Ohtake, A. Belyaev, and H.-P. Seidel, "Ridge-Valley Lines on Meshes via Implicit Surface Fitting," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 609-612, 2004.

[24] G. Medioni, M.-S. Lee, and C.-K. Tang, *Computational Framework for Segmentation and Grouping.* Elsevier Science Inc., 2000.

[25] S. Wang, T. Hou, Z. Su, and H. Qin, "Diffusion Tensor Weighted Harmonic Fields for Feature Classification," *Proc. Conf. Pacific Graphics,* pp. 93-98, 2011.

[26] M. Mortara, G. Patané, M. Spagnuolo, B. Falcidieno, and J. Rossignac, "Blowing Bubbles for Multi-Scale Analysis and Decomposition of Triangle Meshes," *Algorithmica,* vol. 38, no. 1, pp. 227-248, 2003.

[27] C. Rssl, L. Kobbelt, H.-P. Seidel, and I. Stadtwald, "Extraction of Feature Lines on Triangulated Surfaces Using Morphological Operators," *Proc. Int'l Symp. Smart Graphics,* pp. 71-75, 2000.

[28] Y. Lai, Q. Zhou, S. Hu, J. Wallner, and H. Pottmann, "Robust Feature Classification and Editing," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 1, pp. 34-45, Jan./Feb. 2007.

[29] M. Sunkel, S. Jansen, M. Wand, E. Eisemann, and H.-P. Seidel, "Learning Line Features in 3D Geometry," *Computer Graphics Forum,* vol. 30, no. 2, pp. 267-276, 2011.

[30] L.Di Angelo and P.Di Stefano, "C1 Continuities Detection in Triangular Meshes," *Computer-Aided Design,* vol. 42, no. 9, pp. 828-839, 2010.

[31] S. Katz and A. Tal, "Hierarchical Mesh Decomposition Using Fuzzy Clustering and Cuts," *ACM Trans. Graphics,* vol. 22, no. 3, pp. 954-961, 2003.

[32] Y. Lai, Q. Zhou, S. Hu, and R.R. Martin, "Feature Sensitive Mesh Segmentation," *Proc. ACM Solid and Physical Modeling Symp.,* pp. 17-25, 2006,

[33] O. Sorkine, D. Cohen-Or, R. Goldenthal, and D. Lischinski, "Bounded-Distortion Piecewise Mesh Parameterization," *Proc. Conf. Visualization,* pp. 355-362, 2002.

[34] Z. Ji, L. Liu, Z. Chen, and G. Wang, "Easy Mesh Cutting," *Computer Graphics Forum,* vol. 25, no. 3, pp. 283-291, 2006.

[35] Y. Zheng and C.-L. Tai, "Mesh Decomposition with Cross-Boundary Brushes," *Computer Graphics Forum,* vol. 29, no. 2, pp. 527-535, 2010.

[36] R. Liu and H. Zhang, "Segmentation of 3D Meshes Through Spectral Clustering," *Proc. 12th Pacific Conf. Computer Graphics and Applications,* pp. 298-305, 2004.

[37] H.Y. Seungyong, H. Yamauchi, S. Lee, Y. Lee, Y. Ohtake, and E.B.H. peter Seidel, "Feature Sensitive Mesh Segmentation with Mean Shift," *Proc. Int'l Conf. Shape Modeling,* pp. 236-243, 2005.

[38] C. Xiao and M. Liu, "Efficient Mean-Shift Clustering Using Gaussian Kd-Tree," *Computer Graphics Forum,* vol. 29, no. 7, pp. 2065-2073, 2010.

[39] D. Cohen-Steiner, P. Alliez, and M. Desbrun, "Variational Shape Approximation," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 905-914, 2004.

[40] C.C.L. Wang, "Extracting Manifold and Feature-Enhanced Mesh Surfaces from Binary Volumes," *J. Computing and Information Science in Eng.,* vol. 8, no. 3, 2008.

[41] D.-M. Yan, W. Wang, Y. Liu, and Z. Yang, "Variational Mesh Segmentation via Quadric Surface Fitting," *Computer-Aided Design,* vol. 44, no. 11, pp. 1072-1082, 2012.

[42] J. Zhang, J. Zheng, C. Wu, and J. Cai, "Variational Mesh Decomposition," *ACM Trans. Graphics,* vol. 31, no. 3, pp. 21:1-21:14, 2012.

[43] O. Kin-Chung Au, Y. Zheng, M. Chen, P. Xu, and C.-L. Tai, "Mesh Segmentation with Concavity-Aware Fields," *IEEE Trans. Visualization and Computer Graphics,* vol. 18, no. 7, pp. 1125-1134, July 2012.

[44] A. Shamir, "A Survey on Mesh Segmentation Techniques," *Computer Graphics Forum,* vol. 27, no. 6, pp. 1539-1556, 2008.

[45] H. Benhabiles, J.-P. Vandeborre, G. Lavoué, and M. Daoudi, "A Comparative Study of Existing Metrics for 3D-Mesh Segmentation Evaluation," *The Visual Computer,* vol. 26, no. 12, pp. 1451-1466, 2010.

[46] R.A. Horn and C.R. Johnson, *Matrix Analysis.* Cambridge Univ. Press, 1985.

[47] S. Wang, T. Hou, Z. Su, and H. Qin, "Multi-Scale Anisotropic Heat Diffusion Based on Normal-Driven Shape Representation," *The Visual Computer,* vol. 27, nos. 6-8, pp. 429-439, 2011.

[48] K. Xu, H. Zhang, D. Cohen-Or, and Y. Xiong, "Dynamic Harmonic Fields for Surface Processing," *Computers and Graphics,* vol. 33, no. 3, pp. 391-398, 2009.

[49] T.A. Davis and W.W. Hager, "Dynamic Supernodes in Sparse Cholesky Update/Downdate and Triangular Solves," *ACM Trans. Math. Software,* vol. 35, no. 4, pp. 1-23, 2009.

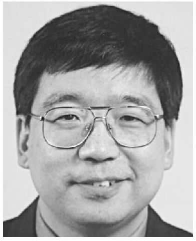
[50] K. Demarsin, D. Vanderstraeten, T. Volodine, and D. Roose, "Detection of Closed Sharp Edges in Point Clouds Using Normal Estimation and Graph Theory," *Computer-Aided Design,* vol. 39, no. 4, pp. 276-283, 2007.

Shengfa Wang received the BS and PhD degrees in computational mathematics from the Dalian University of Technology. He is a postdoctoral of Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, and works in the School of Software Technology, Dalian University of Technology. His research interests include computer graphics, diffusion geometry, and differential geometry processing and analysis.

Tingbo Hou received the BS degree from the University of Science and Technology of China, the ME degree from the Chinese Academy of Sciences, and the PhD degree in computer science from Stony Brook University. His research interests include computer graphics, diffusion geometry and differential geometry processing and analysis.

Shuai Li received the PhD degree in computer science from Beihang University. He is currently a faculty member at the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. His research interests include computer graphics, real-time and realistic rendering, physics-based modeling and simulation, and medical image processing.

Zhixun Su received the BS degree in mathematics from Jilin University, the MS degree in computer science from Nankai University, and the PhD degree in computational mathematics from the Dalian University of Technology. He is a professor at the School of Mathematical Sciences and the director in the Key Lab of Computational Geometry, Graphics and Images, Dalian University of Technology. His research interests include computational geometry, computer graphics, image processing, and computer vision.



Hong Qin received the BS and MS degrees in computer science from Peking University. He received the PhD degree in computer science from the University of Toronto. He is a professor of computer science in the Department of Computer Science, Stony Brook University. His research interests include geometric and solid modeling, graphics, physics-based modeling and simulation, computer-aided geometric design, visualization, and scientific computing. He is a senior member of the IEEE.