Graphical Models 119 (2022) 101121

Contents lists available at ScienceDirect

Graphical Models

journal homepage: www.elsevier.com/locate/gmod

Multi-scale and multi-level shape descriptor learning via a hybrid fusion network

Xinwei Huang [a], Nannan Li [b], Qing Xia [a,c], Shuai Li [a], Aimin Hao [a], Hong Qin [*,d]

[a] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China
[b] School of Information Science and Technology, Dalian Maritime University, China
[c] SenseTime Research, China
[d] Department of Computer Science, Stony Brook University (SUNY at Stony Brook), New York 11794-2424, USA

ARTICLE INFO

Keywords:
3D shape descriptors
Deep learning
Fusion network
Hand-crafted features
Partial retrieval
Shape recognition

ABSTRACT

Discriminative and informative 3D shape descriptors are of fundamental significance to computer graphics applications, especially in the fields of geometry modeling and shape analysis. 3D shape descriptors, which reveal extrinsic/intrinsic properties of 3D shapes, have been well studied for decades and proved to be useful and effective in various analysis and synthesis tasks. Nonetheless, existing descriptors are mainly founded upon certain local differential attributes or global shape spectra, and certain combinations of both types. Conventional descriptors are typically customized for specific tasks with priori domain knowledge, which severely prevents their applications from widespread use. Recently, neural networks, benefiting from their powerful data-driven capability for general feature extraction from raw data without any domain knowledge, have achieved great success in many areas including shape analysis. In this paper, we present a novel hybrid fusion network (HFN) that learns multi-scale and multi-level shape representations via uniformly integrating a traditional region-based descriptor with modern neural networks. On one hand, we exploit the spectral graph wavelets (SGWs) to extract the shapes' local-to-global features. On the other hand, the shapes are fed into a convolutional neural network to generate multi-level features simultaneously. Then a hierarchical fusion network learns a general and unified representation from these two different types of features which capture multi-scale and multi-level properties of the underlying shapes. Extensive experiments and comprehensive comparisons demonstrate our HFN can achieve better performance in common shape analysis tasks, such as shape retrieval and recognition, and the learned hybrid descriptor is robust, informative, and discriminative with more potential for widespread applications.

1. Introduction and motivation

With the rapid advances of the 3D scanning technique and low-cost 3D acquisition devices, the amount of 3D models grows quickly. More and more graphics and vision tasks call for the accurate analysis of these models such as retrieval, classification, segmentation and so on [1–3]. Accordingly, constructing a discriminative and robust multi-scale descriptor is becoming more and more important for those down-stream applications.

Over the past decades, various descriptors have been proposed to solve different kinds of tasks and these descriptors can be mainly divided into two categories: hand-crafted descriptors and learning-based descriptors. Hand-crafted descriptors principally focus on certain differential or geometric nature of 3D shapes and represent a shape using the statistics of corresponding specially designed signal/function distribution [4–6]. Hand-crafted descriptors have been widely used due to the efficient calculation and strong robustness. However, these descriptors are often limited by the discriminative power of manually defined representation and hand-tuned/task-specific parameters. Different from hand-crafted descriptors, learning-based descriptors can bring better performance by exploiting machine learning technique to learn features from shapes in an automatic and more effective way, especially for deep neural networks which have already achieved great success in many related fields [7–9]. With the elaborately designed network structure, deep learning based methods can learn complicated mappings while requiring minimal domain knowledge. However, due to the different data types, the networks designed for images generally can not be implemented on 3D models directly. Besides, deep learning based

* Corresponding author.
E-mail address: qin@cs.stonybrook.edu (H. Qin).

https://doi.org/10.1016/j.gmod.2021.101121
Received 12 October 2020; Received in revised form 3 August 2021; Accepted 12 November 2021
Available online 22 November 2021

methods call for a large number of samples to train, and the network can only process the specific data category in some cases. Based on different types of data (mesh, volumetric grid, point cloud, and multi-view), many methods are proposed to adapt CNN designed for 2D images to 3D shape models.

While descriptors, including both hand-crafted features and deep learning based descriptors, have achieved reasonable performance in various scenarios, there are few works concerning about the fusion of hand-crafted features within a deep learning framework. In this paper, we propose a novel multi-scale and multi-level descriptor on 3D shapes represented as triangular meshes, which integrates hand-crafted descriptors into a deep network structure. With the combination, the fused descriptor is more informative and discriminative in model recognition, shape retrieval, and many other applications. In general, the contribution of our paper can be summarized as follows,

- We propose a hierarchical neural network structure designed for mesh data. Faces are regarded as the base unit of the input for the network. With the multi-scale design, robust local structures are captured to form a novel local descriptor.
- We propose a framework that fuses hand-crafted features and deep learning based features. Based on the framework, a more informative and discriminative fused descriptor is further proposed.
- Experiment results show that our proposed descriptor improves the performance on both shape retrieval and model recognition. The fusion framework can exploit both hand-crafted and deep learning based descriptors efficiently.

The remainder of this paper is organized as follows: Section 2 introduces the related work. Section 3 gives the details of the fusion descriptors. Section 4 demonstrates the experiment results. Section 5 concludes the paper.

2. Related work

In this section, we first review the existing hand-crafted descriptors. Then deep learning based descriptors are generally introduced based on the data type.

2.1. Hand-crafted descriptors

Various hand-crafted descriptors have been proposed for different applications and data structure. These hand-crafted descriptors can be roughly divided into two categories: extrinsic descriptors and intrinsic descriptors. Extrinsic descriptors, such as Spin images (SI) [10], 3D shape context (3DSC) [11], random forest based descriptors [12], and distance-based descriptors [13], are usually collected from the statistic information of coordinates or angles. Distances between random pairs [6], Zernike Moments [14] and Shape Diameter Function [15] are also used in various applications. Intrinsic descriptors exploit spectral descriptors on manifolds, which are invariant to isometric deformation. Classic works include Intrinsic Shape Context (ISC) descriptor [16], Shape-DNA [17], Local-to-Global Shape Feature [18], global point signatures [19], the heat kernel signatures [4], scale-invariant HKS [5], wave kernel signatures [20] and Local Point Signature [21], etc. Hand-crafted descriptors efficiently exploit statistic and distribution information. The training dataset and learning process are often not needed due to the unsupervised learning strategy. Though efficient and robust, hand-crafted descriptors often face the problem of the low discriminative power of manually designed features and hand-tuned parameters.

2.2. Deep learning based descriptors

Recently, with the rapid growth of deep convolutional neural networks, many deep learning based descriptors have been proposed. Meanwhile, the amount of 3D models grows quickly due to the advanced 3D scanning technique and devices. Large-scale 3D model datasets are exploited by the extraordinary power of the neural networks and leading performance has been achieved. Due to the different data representation, the conventional CNN based methods for 2D images can not be used on 3D model shapes directly. Accordingly, deep learning based methods can be roughly divided into four categories: voxel-based methods, point-based methods, view-based methods, and mesh-based methods.

Voxel-based methods 3D models represented by volumetric grid use ordered voxels to record the information. It is intuitive to use voxels as a fundamental unit to construct a 3D convolutional neural network. 3D ShapeNets [22] and VoxNet [23] first treat voxels information as a probability distribution of binary variables and extend 2D image convolution into 3D model networks. However, more dimensions lead to much more computation cost. As a result, 3D models need to degenerate into low-resolution samples. These compromising methods cause the sparsity of data and waste of computation. To solve this problem, OctNet [24] and O-CNN [25] hierarchically partition the space using a set of unbalanced octrees to store the voxel information efficiently. Field probing neural networks (FPNN) [26] uses field probing filters to extract features from 3D vector fields.

View-based methods View-based methods recognize 3D shapes from their rendered 2D images. By setting cameras around the model, features are extracted from images of different views. With a multi-view CNN (MVCNN) [7] structure, information from all views is synthesized into a single compact 3D shape descriptor. Feng et al. [27] focus on the intrinsic hierarchical correlation and discriminability among views. A grouping module and view pooling are proposed. Views of each shape are first grouped into different clusters to gain associated weights.

Point-based methods Point cloud is an irregular format to represent 3D models. To solve the problem, Qi et al. [8] exploit a symmetry function for the unordered input. Then PointNet++ [28] further proposes a hierarchical neural network to capture local structure information. By recursively applying PointNet on a nested partitioning of the input point set, multi-scale information is obtained according to the local point densities. Klokov et al. [29] construct a top-down kd-tree to solve the irregularity. Points are ranked and classified by the coordinate axis. EdgeConv [30] defines an edge feature to better capture local geometric features while still maintaining permutation invariance. So-Net [31] models the spatial distribution of point cloud by building a self-organizing Map.

Mesh-based methods Mesh data consists of different kinds of information such as vertices, edges, and faces. The irregularity and complexity make it difficult to use deep learning methods on a mesh directly. To solve the problem, Geodesic Convolutional Neural Networks (GCNN) [32] proposes a generalization of the convolutional networks on non-Euclidean manifolds. EdgeNet [33] and MeshCNN [34] use edges as the input unit of the network. A pooling layer is used to solve the irregularity of edges. MeshNet [9] regards faces as the unit and captures the spatial feature and structural feature separately. Wang et al. [35] develop a method for extracting short shape descriptors from lengthy descriptors by exploiting the capabilities of the dimensionality reduction methods and the deep convolutional residual network.

Fusion methods There are a few existing fusion methods exploiting multiple data types. In PVNet [36], global features are first taken from multi-view data and then projected into the subspace of point cloud features with an embedding network. FusionNet [37] proposes a network structure that uses both volumetric and multi-view data. Although these fusion methods utilize information from different data types, there are few methods focusing on how to embed hand-crafted descriptors into a deep learning network structure.

3. Our method

We propose a multi-scale fusion network that integrates hand-crafted

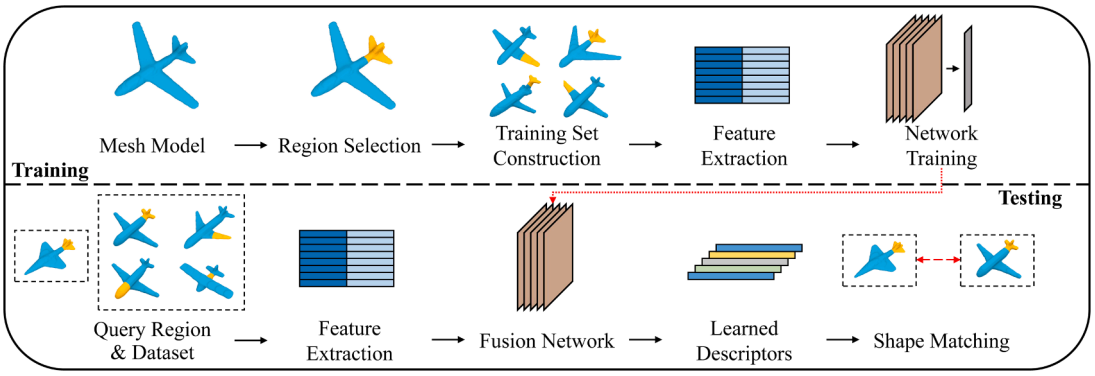

Fig. 1. The pipeline of our fusion network.

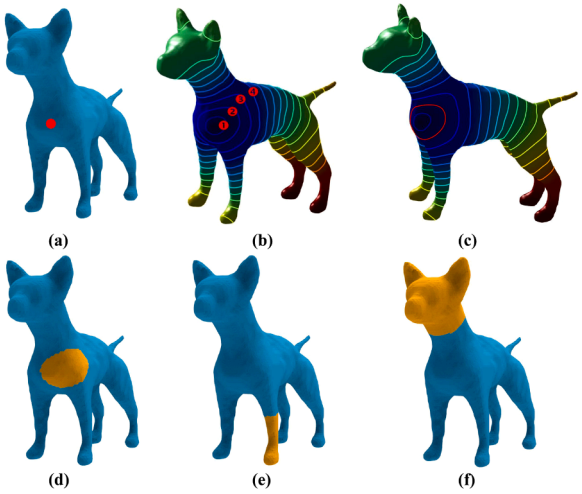

Fig. 2. Examples of the region selection. First in (a), an interest point is selected. Then (b) shows the corresponding contours based on the bi-harmonic distance. (c) shows an example of choosing No. 2 contour in (b). (d), (e), (f) show more examples with different scales and regions.

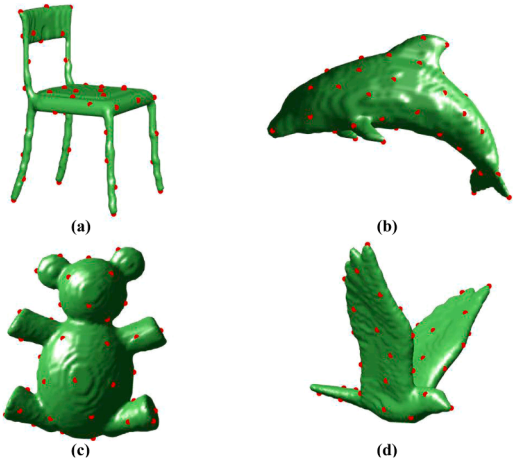

Fig. 3. Examples of the point selection based on the farthest-point-sampling method on a chair (a) and a dolphin (b) model. It shows that this sampling strategy can choose the representative endpoints uniformly.

descriptors and deep learning based descriptors together. The network captures local descriptors of user-specified partial regions on mesh data. The pipeline of our approach is shown in Fig. 1. For the training process, query model and its specific region are selected. Then the training set is constructed and raw features are extracted. The features are exploited to train the fusion network. In the testing process, raw features are fed into the well-trained network to generate new learned descriptors, which are used for further applications. In this section, we first introduce the definition and choice of the partial regions. Then we give the details of the local hand-crafted descriptors. Then we introduce the multi-scale fusion network.

3.1. Region selection

Mesh data consists of the vertices, edges, and faces of 3D models. Compared with other data formats, mesh data contains more detailed information about the models structure, topology, and formation. These information can be extracted either from specific hand-crafted descriptors or raw features by deep learning network. Besides, unlike point cloud data, the local structure of mesh data is accurate due to the edges which represent the explicit connection relationship between vertices. Inspired by Li et al. [18], we exploit bi-harmonic distance field [38] to define the user-specified partial regions.

Given a 3D mesh represented as $\mathbf{M} = (\mathbf{V}, \mathbf{E}, \mathbf{F})$, where $\mathbf{V} = \{v_1, v_2, \ldots, v_n\}$, $\mathbf{E}$, $\mathbf{F}$ denotes vertices, edges and faces, respectively, to choose an interest partial region, we apply the basic approach in Li et al. [18], as shown in Fig. 2. One needs to decide an interest point v_s and a contour scope. In general, any vertex can be chosen as the interest point and the interest point will be the relative center of the partial region. For simplicity, several representative points are chosen with the farthest-point-sampling strategy [39]. This strategy ensures the uniform distribution of the candidate points and the inclusion of the end points. Examples are shown in Fig. 3. After the interest point v_s is selected, the contour scope can be decided with bi-harmonic distance. According to [38], the bi-harmonic distance between any given vertices v_i and v_j can be calculated as following:

$$D(i,j)^2 = \sum_{k=1}^{m} \frac{(\chi_k(i) - \chi_k(j))^2}{\lambda_k^2}, \tag{1}$$

where λ_k is the smallest m non-zero eigenvalues and $\chi_k(\cdot)$ is the corresponding eigenfunction derived from the discrete Laplace–Beltrami operator with cotangent formula [40]. Based on Eq. (1), for an interest point v_s, the distance between v_s and other points is calculated and a diffusion field around v_s is formed. With this diffusion field, we can set multiple contours to decide regions with different scales sufficiently. The contours are set with points located on the edges of the mesh model. With different scales, the contours can efficiently reflect the local-to-global structure of v_s. A normalization step is taken on models with different sizes using a unit box. Then the number of contours is empirically set as $[\max(D(s,\cdot))/0.05]$, where $D(s,\cdot)$ denotes the bi-harmonic distance between v_s and other vertices.

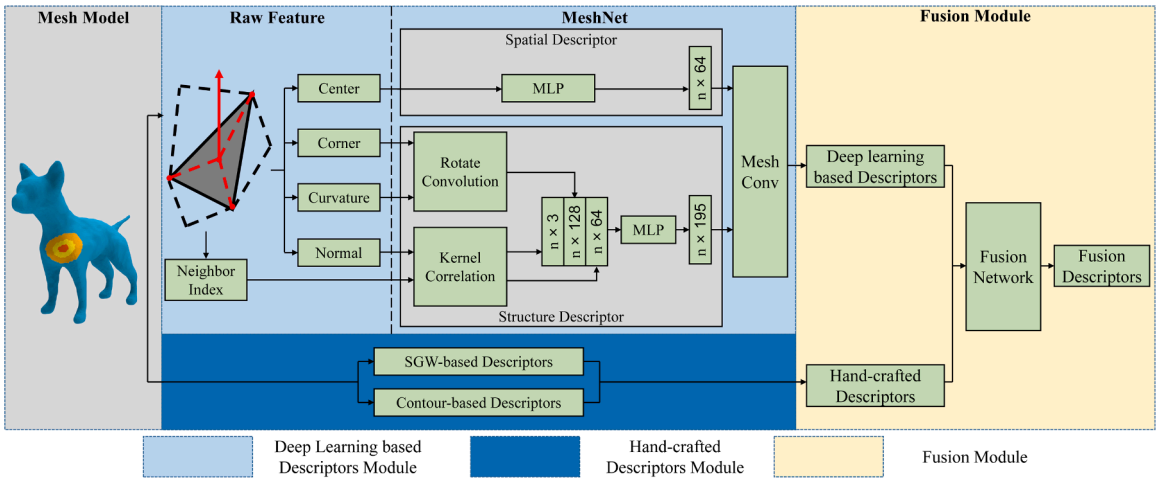

Fig. 4. An overview of our proposed fusion network structure. Given a triangulated mesh, we first choose an interest region. The region is further divided into multi-scale subsets. Then raw features are fed into a modified MeshNet, while SGW-based and contour-based hand-crafted features are constructed at the same time. Both features are integrated into a unified fusion descriptor via the fusion module.

3.2. Hand-crafted descriptors

We use Local-to-Global Shape Feature [18] as our hand-crafted descriptors due to its strong ability to capture local information. The feature is exploited in our fusion net and a brief introduction is given in this section. The Local-to-Global Shape Feature contains two parts: spectral graph wavelet (SGW)-based description and contour-based multi-scale statistics. Both parts are fitted into the aforementioned Laplacian–Beltrami eigenfunctions and the corresponding contours. SGW constructs wavelets on weighted graphs and the scale can be implemented in the spectral domain of the graph Laplacian. SGW with a generating kernel g can be expressed as the following bivariate kernel functions,

$$\begin{aligned}\psi_t(i,j) &= \sum_{k=0}^{n-1} \mathrm{g}(t\lambda_k)\chi_k(i)\chi_k(j),\\ \psi_{t,i}(\cdot) &= \psi_t(\cdot,\cdot) = \sum_{k=0}^{n-1} \mathrm{g}(t\lambda_k)\chi_k(i)\chi_k(\cdot).\end{aligned} \tag{2}$$

where g is the real-valued SGWs generating kernel, which satisfies $\mathrm{g}(0) = 0$ and $\lim_{x\to\infty}\mathrm{g}(x) = 0$, $\psi_{t,i}$ is the ith row of $\psi_t(\cdot,\cdot)$. Then for an interest point v_s and a vectorvalued function f, we can get the wavelet coefficients at scale t in frequency domain as:

$$\begin{aligned}W_f(t,i) &= \langle \psi_{t,i}, f\rangle = \sum_{l=0}^{n-1} g(t\lambda_l)\widehat{f}(l)\chi_l(i),\\ \widehat{f}(l) &= \langle \chi_l, f\rangle = \sum_{i=0}^{n-1}\chi_l(i)f(i).\end{aligned} \tag{3}$$

We follows the setting of scale t, generating kernel g and signal function f in Li et al. [18], Hammond et al. [41]. t is selected to be distributed logarithmically in $[2/\lambda_{\max}, 40/\lambda_{\max}]$. g is set as:

$$\mathrm{g}(x) = \begin{cases} x^2 & (x<1)\\ -5+11x-6x^2+x^3 & (1\le x\le 2)\\ 4x^{-2} & (x>2)\end{cases}. \tag{4}$$

And f can be set as any signal function, such as mean curvature and characterizing detailed distortions, according to different applications.

The second part of the Local-to-Global Shape Feature is based on the contours of bi-harmonic field. The perimeters of contours with different scales are calculated because they encode rich local-to-global geometric variation. The perimeters are expressed as $\{p^{c_1}, p^{c_2}, \ldots, p^{c_L}\}$, with c_i denotes ith contour at different scales. Besides, the Euclidean distances between contour points and their barycenter are exploited to express the inner structure of the partial region. To be detailed, the Euclidean distances are first calculated and then separated into M bins uniformly. The probability distribution of the distances is evaluated as:

$$ds^{c_i} = \left[\frac{num(b_1)}{num(c_i)}, \frac{num(b_2)}{num(c_i)}, \ldots, \frac{num(b_M)}{num(c_i)}\right], \tag{5}$$

where $num(b_i)$ denotes the number of points that falls into the ith bin. $num(c_i)$ denotes the total number of points on the ith contour. Finally, the aforementioned features are concatenated together to form the hand-crafted descriptors.

3.3. Overall network structure

This section introduces the network framework that fuses the hand-crafted features and deep learning based descriptors together. The overall architecture of the fusion network is illustrated in Fig. 4. First, an interest point and corresponding contour are selected based on the method in Section 3.1. The interest point and contour decide about the partial region, which is regarded as the input unit of the fusion network. With a specific region, we can easily get the vertices, edges, and faces. It should be noted that only faces that are completely located in the region are selected. Then we further subdivide the region into thinner bands with smaller contours. The contours are also set based on bi-harmonic distance as those in Section 3.1 and the number of contours is empirically set as $[\max(D(s,\cdot))/0.025]$. The dense contours divide the feature region into several concentric parts with different scope contours. We treat these concentric parts as the different views of multi-view based methods. The parts with different radii can reflect the local geometry structure with different scales. Both hand-crafted features and deep learning based descriptors are extracted separately with a parameter shared network structure. The deep learning based descriptors are extracted with modified MeshNet, whose details are given in the next subsection. The features of different concentric parts are then concatenated and stacked. After a normalization step, the concatenation feature is passed through a fusion network to generate the fusion descriptors for different applications.

3.4. Modified MeshNet

Inspired by Feng et al. [9], we modify the framework of MeshNet to extract deep learning based descriptors that are suitable for our fusion network. Although mesh data has extensive information to describe 3D

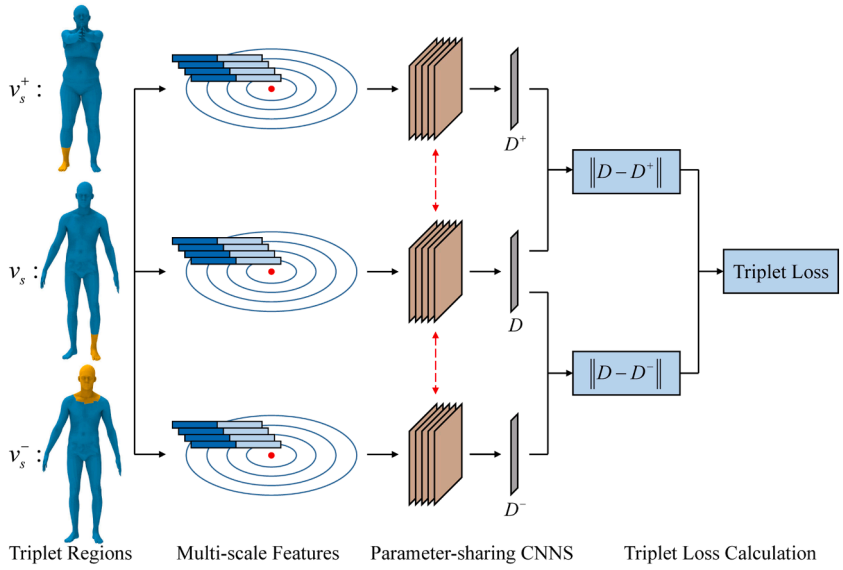

Fig. 5. The overview of the fusion module. Multi-scale deep learning based and hand-crafted descriptors are first concatenated and stacked. Then features are passed through a triplet network, which consists of three parameter-sharing CNNs. A triplet loss is calculated. By minimizing the triplet loss, the fusion module can generate fusion descriptors that ensure distances between matching points keep small, while distances between mismatched points remain large.

shapes, it should face the problem of irregularity when mesh data is passed through a convolutional neural network. To solve the problem, faces are regarded as the input unit and a symmetry function is implemented. The overall structure of the modified MeshNet is shown in Fig. 4. First, the raw features of faces are extracted. Besides center, corner, normal and neighbor index which are used in Feng et al. [9], we further extract the Gaussian curvature to capture more informative local features. The Gaussian curvature is a central property of a geometric surface, which can reflect the local shape information. For a given point on a smooth surface, the Gaussian curvature can be easily calculated as the product of the two principal curvatures. While for a vertex on a triangulated mesh, the Gaussian curvature can be approximately conducted by angles and areas. Given a vertex on the triangulated mesh, we can find b triangles around that share the vertex. For each triangle, we use $\theta_i, i \in \{1, 2, \ldots, b\}$ to denote the degree of the angle at the vertex. Then the Gaussian curvature κ is computed as follows:

$$\kappa = \frac{1}{\frac{1}{3}\sum_{i=1}^{b} A_i}\left(2\pi - \sum\nolimits_{i=1}^{b}\theta_i\right), \tag{6}$$

where A_i is the area of the i-th triangle. Then for faces in the interest region, the Gaussian curvature of the corresponding three vertices is calculated as one of the inputs of the network. In general, center, normal, neighbor index, and Gaussian curvature are 3-dimensional features, which denote the coordinate of the center point, the unit normal vector, the indexes of the connected faces, the Gaussian curvature of vertices, respectively. The corner is a 9-dimensional feature that denotes vectors from the center point to three vertices.

After the raw features of faces are extracted, we apply the network structure of MeshNet and divide the features into two modules: the spatial descriptor module and the structural descriptor module. The spatial descriptor module exploits the spatial information that reflects the location where the faces are. The structural descriptor module processes local structure and shape information to analyze how the local region looks like. The center value of faces is passed through the spatial descriptor module. A shared Multilayer Perceptron (MLP) similar to [8] is implemented to generate a 64-dimension spatial feature. Other features, the corner, normal, neighbor index, and Gaussian curvature, are passed through the structure descriptor module. First, the normal and neighbor index are processed with j learnable kernels. The kernel correlation is as follows:

$$KC(i,j) = \frac{1}{|U_i||W_j|}\sum_{u\in U_i}\sum_{w\in W_j} K_\sigma(u,w), K_\sigma(u,w) = \exp\left(-\frac{\| u - w \|^2}{2\sigma^2}\right). \tag{7}$$

where U_i denotes the i-th face and its corresponding three neighbors, W_j denotes the set of j learnable kernels. The neighbor index is not directly used in the calculation, but provides the neighbor information of faces. The above kernel correlation exploits the Euclidean distance between normals and kernels to generate kernels that can reflect the common distributions of the 3D shape. Then the j-dimensional features are converted into a 64-dimensional feature with an MLP. Then for the corner and Gaussian curvature feature, which contains three values or vectors for each face, the problem of irregularity is solved by a rotation operation. Take Gaussian curvature as an example. Supposing the Gaussian curvature of a face to be $[gc_1, gc_2, gc_3]$, the output can be calculated as follows:

$$g'\left(\frac{1}{3}(f'(gc_1, gc_2) + f'(gc_2, gc_3) + f'(gc_1, gc_3))\right), \tag{8}$$

where g' and f' can be any valid function. In experiments, $f'(\cdot,\cdot)$ and $g'(\cdot)$ are implemented by fully-connected layers $(32, 32)$ and $(64, 64)$. With a shared MLP, a 64-dimensional feature is generated for the corner and Gaussian curvature separately.

After all the raw features are processed with the corresponding module, the generated spatial and structure features are then passed through a Mesh Convolution module, in which features of neighboring faces are aggregated together to capture more local features. The details can be found in Feng et al. [9]. With the modified MeshNet, a 1024-dimensional feature is finally extracted for a given interest region. The modified MeshNet is pre-trained and fine-tuned on the benchmark ModelNet40 [22] dataset.

3.5. Fusion module

This subsection introduces the fusion module of our network. For a given interest region, both hand-crafted features and deep learning based features are extracted with different scales. The hand-crafted features and deep learning based features with one certain scale are first normalized separately and then concatenated into one united feature. Then the concatenated features of different scales are stacked

and passed through the fusion module. As shown in Fig. 5, the fusion module is designed as a triplet CNNs structure. For every given interest point v_s, a good matching point v_s^+ and a mismatching point v_s^- are chosen to form a triplet (v_s, v_s^+, v_s^-). Then the interest regions are selected, and the corresponding concatenated features of the triplet are extracted. The features are fed into deep triplet CNNs [42] to generate learned descriptors. The deep triplet CNNs consist of three identical CNNs which share the same structures and parameters. A triplet loss is defined based on the learned descriptors:

$$\begin{aligned} d^+ &= \| D(v_s) - D(v_s^+) \|_2, \\ d^- &= \| D(v_s) - D(v_s^-) \|_2, \\ L &= \sum_{v=1}^{V} \max(0, d^+(v) - d^-(v) + \alpha), \end{aligned} \tag{9}$$

where d^+ and d^- denotes the distance between v_s^+ and v_s^- against v_s, respectively. By minimizing the triplet loss L, which means $d^+ \to 0$ and $d^- \to \alpha$, the learned descriptors can ensure that similar parts of different models are projected into close feature space, while distances of features between different parts remain large.

During the training process, a large number of triplets are selected to train the triplet CNNs. First, for every model in the dataset, points are selected by farthest point sampling (FPS) strategy. Second, for each point, the corresponding points on other models are selected as good matching points. Besides, for each point, points that lie in the geodesic ball of radius r are also regarded as good matching points. Third, those points that lie outside the geodesic ball of radius $R (r < R)$ are regarded as mismatching points. Instead of choosing mismatching points randomly, we pick half of them from the parts lying between radii R and $4R$. Another half are picked from the points lies in the scope farther than $4R$. With this setting of the triplets, the trained triplet CNNs can extract discriminative and robust descriptors automatically. The concatenated features are transformed into a more representative fusion descriptor. During the test process, concatenated features are fed into the trained CNN to generate a final descriptor. Then the Euclidean distance in the new learned descriptor space is used for various applications.

4. Experiments, applications, and evaluations

In this section, extensive experiments and comparisons are implemented to demonstrate our fusion descriptors. The implementation details and performance are introduced.

4.1. Implementation details

For the parameters in hand-crafted descriptors and deep learning based descriptors, we apply the basic setting in Feng et al. [9], Li et al. [18]. When constructing the bi-harmonic distance matrix, the number of first no-zero eigenvalues are set as 300 for each model. For the contour-based part of the hand-crafted descriptors, the number of wavelets time scales and bins in distance distribution is set as 5 and 10, respectively. The experiments are conducted on McGill 3D shape benchmark [43] and FAUST [44]. McGill database consists of 19 categories models and in each category, there are typically 20 to 30 3D models. FAUST is another well-known 3D benchmark dataset in 3D shape corresponding and retrieval, which contains 100 human models from 10 categories with different figures and poses. Before the experiments, the models are translated into the geometric center and normalized. The Gaussian curvature, corner, and normal vector are computed for each vertex and face. The bi-harmonic distance distribution is also calculated for each mesh model. All of these computation needs to be done only once. The MeshNet module is pre-trained on the ModelNet40 dataset containing 12,311 models from 40 categories. Interest points and regions are randomly selected in the dataset. Then their corresponding matching regions in the same or different mesh models are annotated. It should be noticed that the matching regions are not required to be exactly the same due to the differences between models. The interest points and its neighbors are treated as good matching points. The fusion module is tuned by these pairs of regions with the triplet CNN structure. For the testing, the features of query regions and candidate regions are computed first. Then similarity between query regions and candidate regions is measured and ranked with L2 norms. The experiments are conducted on a computer with Intel Core i7-9800X 3.80 GHz with 16 GB RAM.

Table 1
Performance comparison of different detection methods on McGill dataset.

Method	Ant head (30)		Plane tail (20)		Human leg (20)	
	20	40	15	30	30	60
Ideal	100%	75.0%	100	66.7%	100	66.7%
D2 [6]	30.0%	32.5%	26.7%	23.3%	40.0%	41.7%
CF [45]	35.0%	37.5%	53.3%	56.7%	46.7%	48.3%
ZM [14]	50.0%	42.5%	53.3%	53.3%	46.7%	50.0%
SDF [15]	45.0%	57.5%	46.7%	50.0%	50.0%	58.3%
PS [46]	80.0%	62.5%	46.7%	43.3%	73.3%	60.0%
L-G [18]	90.0%	65.0%	80.0%	60.0%	86.7%	63.3%
MeshNet [9]	45.0%	37.5%	46.7%	50.0%	56.7%	51.7%
M-MeshNet	60.0%	45.0%	60.0%	53.3%	66.7%	55.0%
Ours	**95.0%**	**67.5%**	**80.0%**	**63.3%**	**93.3%**	**63.3%**

4.2. Application: partial retrieval

We compare our fusion descriptors with several existing methods of partial retrieval. Here partial retrieval means the specific application that aims to find the matching regions of a user-specified feature region. First, a query region is selected with a specific interest point and contour. Then candidate regions are computed from the dataset. Our fusion features are then extracted and ranked to find the most similar regions.

Due to the special application, we compare our descriptors with the following existing descriptors: D2 shape Distribution (D2) [6], Conformal Factors (CF) [45], Zernike Moments based signature (ZM) [14], Local SDF Signature (SDF) [15], Patch Spectral Geometric Features (PS) [46], MeshNet and Local-to-Global Shape Feature [18]. The first five descriptors are popular existing methods based on region description and comparable performance has been achieved in shape retrieval and conventional partial matching. D2, CF, and SDF exploit the statistics of points distance, conformal geometric factors, and shape diameter function as descriptors, respectively. ZM computes the local shape signature based on the transformation of Zernike Moments. PS shows the normalized spectra of patch spectrum decomposition. The last two descriptors, MeshNet and Local-to-Global Shape Feature, is the deep learning based and hand-crafted features we used in our fusion network, respectively. Besides, M-MeshNet represents the modified MeshNet with Gaussian curvature. We demonstrate these descriptors here as an ablation study to show the comparative performance of our fusion descriptor. To show the retrieval performance intuitively, some meaningful parts of different models are selected artificially, such as plane tail, ant head, etc.

Table 1 shows the retrieval performance on the McGill dataset. Because there are few methods that focus on the user-specified partial retrieval, we take the basic setting in Li et al. [18] for consistency. The experiment results may vary slightly due to the difference of the interest regions. The number of models is annotated behind the category names. For example, for the 'ant head' class, there should be 30 good matching regions. Because there are 30 ant models in the dataset, and for each model a proper part of the head can be selected. When the best 20 retrieved models from each method are checked, there should be 20 good matching regions ideally (100%). And when we check for more retrieved results, such as 40 best retrieved models, the ideal percentage of good matching regions should be 75%. Different parts may have different numbers of good matching results. For the ant head and the

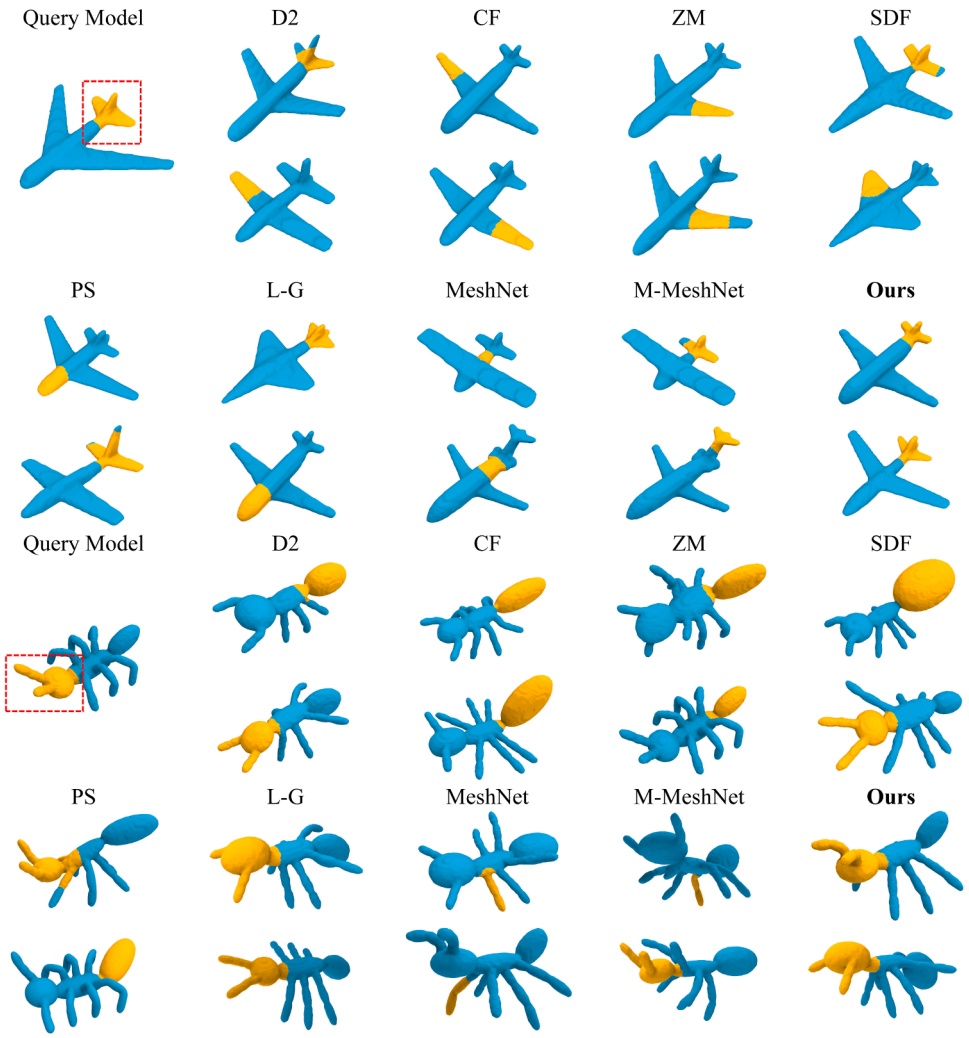

Fig. 6. Examples of partial retrieval on McGill dataset. Results show that our descriptor can retrieve related regions precisely.

plane tail, the number of matching regions is the same as the number of models, as for each model only one matching part exists. While for human leg, the number of matching regions doubles because of the symmetry. The aforementioned features are treated as the input of the triplet CNNs. And for each feature, a trained CNN is obtained and used to extract the final descriptors.

According to the experiment results, our fusion descriptor outperforms other methods in all the categories. Besides, unlike other methods, our fusion descriptor performs stably in different categories. The SGW-based descriptors establish a power hierarchical structure and the contour-based descriptors can capture informative local features. It can be seen from the table that trained CNN based on ModelNet alone performs less satisfied. By taking Gaussian curvature into consideration, the performance of CNN based on modified ModelNet is better due to a more comprehensive knowledge of the local shape information. It should be noticed that the CNN based on Local-to-Global descriptor performs much better than CNNs based on other hand-crafted descriptors. Although the Local-to-Global descriptor is hand-crafted, it is well designed for the partial retrieval task. And it contains the SGWs based descriptors, the statistical information, and structure information, which are representative and discriminative for local parts of a 3D model. Compared with CNN based on hand-crafted or deep learning based features alone, our descriptor performs better due to the fusion strategy. As the input of the fusion module, the concatenated features actually give more weights to the hand-crafted features than the raw features exploited by the ModelNet structure. Then the fusion module exploits both features to generate a more informative and discriminative descriptor. Some retrieval results of different methods are shown in Fig. 6.

Table 2
Performance comparison of different detection methods on FAUST dataset.

Method	D2 [6]	L-G [18]	MeshNet [9]	M-MeshNet	Ours
Head	45%	70%	50%	55%	**80%**
Hand	55%	80%	50%	60%	**90%**
Foot	55%	85%	45%	50%	**85%**

Table 2 shows the retrieval results on the FAUST dataset. Similarly, some meaningful regions are selected as the interest regions (human head, foot, hand, etc.). For the training process, five categories shapes are randomly selected to train the network. The others are regarded as candidate models in the query process. It should be noticed that the models from different categories may vary dramatically due to the different figures. Because our fusion features do not contain the semantic information or segmentation information, the corresponding parts of different shapes will not be treated as good matching results, which is also in accordance with the partial retrieval application in this paper. In the experiments, the accuracy of the best 20 retrieved models are treated as the evaluation criterion. As it is shown in Table 2, it can be seen that our fusion descriptors perform better than other methods in all categories. Fig. 7 shows some retrieval examples.

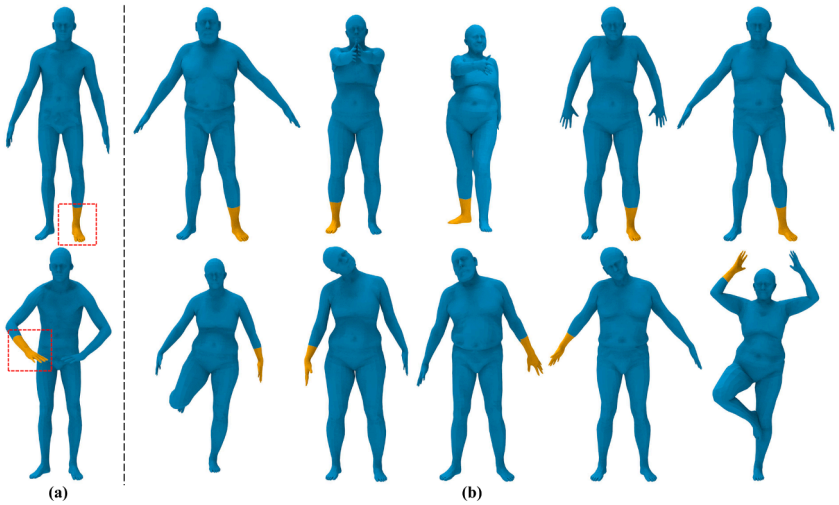

Fig. 7. Partial retrieval results of our approach on FAUST dataset. (a) Query model. (b) Retrieval results.

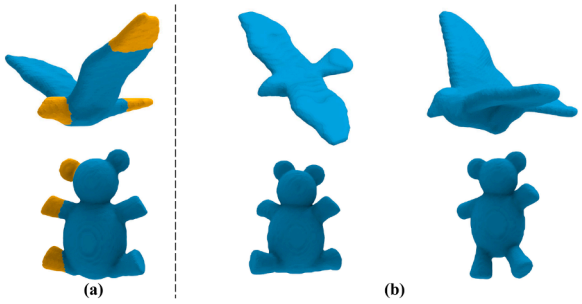

Fig. 8. Examples for model recognition. (a) Feature extraction. (b) Recognition results.

4.3. Application: model recognition

Besides partial retrieval, we utilize our fusion descriptor in the model recognition. For a query model, several key components, such as the wings and tail for a plane, are selected and the corresponding features are extracted. The key components can be selected with human experience or selected easily with the method in Sipiran and Bustos [47]. The sum distances between the selected regions and corresponding parts of candidate models are calculated as the evaluation criteria. The models with small distances are considered to be more likely in accordance with the selected parts. Fig. 8 shows several recognition results. Take the bird model for example, the head, wing, and tail are regarded as three key components. The models with different position and size are retrieved due to the similar partial regions. More components lead to higher retrieval precision. It shows that our descriptor can retrieve matching models without the information of the entire 3D model.

5. Discussion, conclusion, and future work

In this paper, we presented a novel fusion framework, which could exploit both hand-crafted and deep learning based features of 3D shapes towards a more powerful and unified shape description. The raw features from faces of mesh data were passed through a convolutional neural network to generate deep learning based descriptors. At the same time, hand-crafted descriptors were extracted with the powerful SGW-based methods. Then two kinds of descriptors were integrated together, serving as an input to a fusion network. Experiments have confirmed that our fusion framework can achieve better performance compared with the results using hand-crafted descriptors alone or solely relying on feature learning from raw data with a deep network.

Despite the new methodology and more attractive properties that our current shape descriptors have already exhibited, the current research work still has some limitations. For example, we utilized the local hand-crafted descriptor that could only extract features of certain (localized) regions so far, which means that the current fusion framework could not be generalized to the global shape classification task for the entire 3D shape repository. Our immediate future work will seek to exploit other types of shape descriptors, preferably hand-crafted features that are more global and effective and/or more delicate network structure with improved accuracy and performance, in order to build an even more robust and powerful fusion framework to address more technical challenges. Meanwhile, the interpretability of the descriptors is slightly weakened due to the fusion module which contains the triplet CNNs. We also intend to explore other fusion methods which could generate more discriminative and explainable descriptors that could further broaden the current application horizon in shape modeling and computer graphics.

CRediT authorship contribution statement

Xinwei Huang: Conceptualization, Methodology, Validation, Investigation, Writing – original draft. **Nannan Li:** Conceptualization, Writing – review & editing. **Qing Xia:** Writing – review & editing, Methodology. **Shuai Li:** Supervision, Writing – review & editing. **Aimin Hao:** Funding acquisition, Supervision. **Hong Qin:** Writing – review & editing, Supervision, Conceptualization, Methodology.

CRediT authorship contribution statement

Xinwei Huang: Conceptualization, Methodology, Validation, Investigation, Writing – original draft. **Nannan Li:** Conceptualization, Writing – review & editing. **Qing Xia:** Writing – review & editing, Methodology. **Shuai Li:** Supervision, Writing – review & editing. **Aimin Hao:** Funding acquisition, Supervision. **Hong Qin:** Writing – review & editing, Supervision, Conceptualization, Methodology.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is supported in part by National Key R&D Program of

China (No. 2018YFB1700603), National Natural Science Foundation of China (Nos. 61672077 and 61532002), Beijing Natural Science Foundation-Haidian Primitive Innovation Join Fund (L182016). HQ's research has been supported by NSF IIS-1715985 and NSF IIS-1812606.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.gmod.2021.101121

References

[1] A.M. Bronstein, M.M. Bronstein, L.J. Guibas, M. Ovsjanikov, Shape google: geometric words and expressions for invariant shape retrieval, ACM Trans. Graph. (TOG) 30 (1) (2011) 1–20.
[2] O.K.-C. Au, Y. Zheng, M. Chen, P. Xu, C.-L. Tai, Mesh segmentation with concavity-aware fields, IEEE Trans. Vis. Comput. Graph. 18 (7) (2011) 1125–1134.
[3] D. Boscaini, J. Masci, S. Melzi, M.M. Bronstein, U. Castellani, P. Vandergheynst, Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks, Comput. Graph. Forum 34 (5) (2015) 13–23.
[4] J. Sun, M. Ovsjanikov, L. Guibas, A concise and provably informative multi-scale signature based on heat diffusion, Comput. Graph. Forum 28 (5) (2009) 1383–1392.
[5] M.M. Bronstein, I. Kokkinos, Scale-invariant heat kernel signatures for non-rigid shape recognition. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 1704–1711.
[6] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, Shape distributions, ACM Trans. Graph. (TOG) 21 (4) (2002) 807–832.
[7] H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, Multi-view convolutional neural networks for 3D shape recognition. Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 945–953.
[8] C.R. Qi, H. Su, K. Mo, L.J. Guibas, PointNet: deep learning on point sets for 3Dclassification and segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 652–660.
[9] Y. Feng, Y. Feng, H. You, X. Zhao, Y. Gao, MeshNet: mesh neural network for 3D shape representation. Proceedings of the AAAI Conference on Artificial Intelligence vol. 33, 2019, pp. 8279–8286.
[10] A.E. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3D scenes, IEEE Trans. Pattern Anal. Mach. Intell. 21 (5) (1999) 433–449.
[11] A. Frome, D. Huber, R. Kolluri, T. Bülow, J. Malik, Recognizing objects in range data using regional point descriptors. European Conference on Computer Vision, Springer, 2004, pp. 224–237.
[12] Q. Xia, S. Li, H. Qin, A. Hao, Automatic extraction of generic focal features on 3Dshapes via random forest regression analysis of geodesics-in-heat, Comput. Aided Geom. Des. 49 (DEC) (2016) 31–43.
[13] H. Ling, D.W. Jacobs, Shape classification using the inner-distance, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2) (2007) 286–299.
[14] A. Maximo, R. Patro, A. Varshney, R. Farias, A robust and rotationally invariant local surface descriptor with applications to non-local mesh processing, Graph. Models 73 (5) (2011) 231–242.
[15] L. Shapira, S. Shalom, A. Shamir, D. Cohen-Or, H. Zhang, Contextual part analogies in 3D objects, Int. J. Comput. Vis. 89 (2–3) (2010) 309–326.
[16] I. Kokkinos, M.M. Bronstein, R. Litman, A.M. Bronstein, Intrinsic shape context descriptors for deformable shapes. 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 159–166.
[17] M. Reuter, F.-E. Wolter, N. Peinecke, Laplace–Beltrami spectra as 'shape-DNA' of surfaces and solids, Computer-Aided Des. 38 (4) (2006) 342–366.
[18] N. Li, S. Wang, M. Zhong, Z. Su, H. Qin, Generalized local-to-global shape feature detection based on graph wavelets, IEEE Trans. Vis. Comput. Graph. 22 (9) (2015) 2094–2106.
[19] R.M. Rustamov, Laplace–Beltrami eigenfunctions for deformation invariant shape representation. Proceedings of the Fifth Eurographics Symposium on Geometry Processing, Eurographics Association, 2007, pp. 225–233.
[20] M. Aubry, U. Schlickewei, D. Cremers, The wave kernel signature: a quantum mechanical approach to shape analysis. 2011 IEEE International Conference on Computer Vision Workshops (ICCV workshops), IEEE, 2011, pp. 1626–1633.
[21] Y. Wang, J. Guo, D.-M. Yan, K. Wang, X. Zhang, A robust local spectral descriptor for matching non-rigid shapes with incompatible shape structures. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6231–6240.
[22] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3D shapenets: a deep representation for volumetric shapes. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1912–1920.
[23] D. Maturana, S. Scherer, VoxNet: a 3D convolutional neural network for real-time object recognition. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015, pp. 922–928.
[24] G. Riegler, A. Osman Ulusoy, A. Geiger, OctNet: learning deep 3D representations at high resolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3577–3586.
[25] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, X. Tong, O-CNN: octree-based convolutional neural networks for 3D shape analysis, ACM Trans. Graph. (TOG) 36 (4) (2017) 1–11.
[26] Y. Li, S. Pirk, H. Su, C.R. Qi, L.J. Guibas, FPNN: field probing neural networks for 3D data. Advances in Neural Information Processing Systems, 2016, pp. 307–315.
[27] Y. Feng, Z. Zhang, X. Zhao, R. Ji, Y. Gao, GVCNN: group-view convolutional neural networks for 3D shape recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 264–272.
[28] C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet++: deep hierarchical feature learning on point sets in a metric space. Advances in Neural Information Processing Systems, 2017, pp. 5099–5108.
[29] R. Klokov, V. Lempitsky, Escape from cells: deep Kd-networks for the recognition of 3D point cloud models. Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 863–872.
[30] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph CNN for learning on point clouds, ACM Trans. Graph. (TOG) 38 (5) (2019) 1–12.
[31] J. Li, B.M. Chen, G. Hee Lee, SO-Net: self-organizing network for point cloud analysis. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9397–9406.
[32] J. Masci, D. Boscaini, M. Bronstein, P. Vandergheynst, Geodesic convolutional neural networks on Riemannian manifolds. Proceedings of the IEEE International Conference on Computer Vision Workshops, 2015, pp. 37–45.
[33] M. Chen, Q. Zou, C. Wang, L. Liu, EdgeNet: deep metric learning for 3D shapes, Comput. Aided Geom. Des. 72 (JUN) (2019) 19–33.
[34] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, D. Cohen-Or, MeshCNN: a network with an edge, ACM Trans. Graph. (TOG) 38 (4) (2019) 1–12.
[35] Z. Wang, H. Lin, X. Yu, Y.F. Hamza, A dimensional reduction guiding deep learning architecture for 3D shape retrieval, Comput. Graph. 81 (JUN) (2019) 82–91.
[36] H. You, Y. Feng, R. Ji, Y. Gao, PVNet: a joint convolutional network of point cloud and multi-view for 3D shape recognition. Proceedings of the 26th ACM International Conference on Multimedia, 2018, pp. 1310–1318.
[37] V. Hegde, R. Zadeh, FusionNet: 3D object classification using multiple data representations, arXiv preprint arXiv:1607.05695 (2016).
[38] Y. Lipman, R.M. Rustamov, T.A. Funkhouser, Biharmonic distance, ACM Trans. Graph. (TOG) 29 (3) (2010) 1–11.
[39] C. Moenning, N.A. Dodgson, Fast Marching Farthest Point Sampling for Point Clouds and Implicit Surfaces. Technical Report, University of Cambridge, Computer Laboratory, 2003.
[40] M. Meyer, M. Desbrun, P. Schröder, A.H. Barr, Discrete differential-geometry operators for triangulated 2-manifolds. Visualization and Mathematics III, Springer, 2003, pp. 35–57.
[41] D.K. Hammond, P. Vandergheynst, R. Gribonval, Wavelets on graphs via spectral graph theory, Appl. Comput. Harmon. Anal. 30 (2) (2011) 129–150.
[42] M. Chen, C. Wang, H. Qin, Jointly learning shape descriptors and their correspondence via deep triplet CNNs, Comput. Aided Geom. Des. 62 (MAY) (2018) 192–205.
[43] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, S. Dickinson, Retrieving articulated 3-D models using medial surfaces, Mach. Vis. Appl. 19 (4) (2008) 261–275.
[44] F. Bogo, J. Romero, M. Loper, M.J. Black, Faust: dataset and evaluation for 3D mesh registration. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3794–3801.
[45] M. Ben-Chen, C. Gotsman, Characterizing shape using conformal factors. The Seventh Eurographics Workshop on 3D Object Retrieval, 2008, pp. 1–8.
[46] J. Hu, J. Hua, Salient spectral geometric features for shape matching and retrieval, Vis. Comput. 25 (5–7) (2009) 667–675.
[47] I. Sipiran, B. Bustos, Key-component detection on 3D meshes using local features. Proceedings of the 5th Eurographics Conference on 3D Object Retrieval, Eurographics Association, 2012, pp. 25–32.