Context-aware 3D Points of Interest Detection via Spatial Attention Mechanism

ZHENYU SHU, School of Computer and Data Engineering, NingboTech University, China and Ningbo Institute, Zhejiang University, China
LING GAO, School of Computer and Data Engineering, NingboTech University, China
SHUN YI*, School of Computer and Data Engineering, NingboTech University, China and School of Mechanical Engineering, Zhejiang University, China
FANGYU WU, School of Computer and Data Engineering, NingboTech University, China
XIN DING, School of Computer and Data Engineering, NingboTech University, China
TING WAN, School of Computer and Data Engineering, NingboTech University, China
SHIQING XIN, School of Computer Science and Technology, ShanDong University, China

Detecting points of interest is a fundamental problem in 3D shape analysis and can be beneficial to various tasks in multimedia processing. Traditional learning-based detection methods usually rely on each vertex's geometric features to discriminate points of interest from other vertices. Observing that points of interest are related to not only geometric features on themselves but also the geometric features of surrounding vertices, we propose a novel context-aware 3D points of interest detection algorithm by adopting the spatial attention mechanism in this paper. By designing a context attention module, our approach presents a novel deep neural network to simultaneously pay attention to the geometric features of vertices and their local contexts during extracting points of interest. To obtain satisfactory extraction results, our method adaptively assigns different weights to those features in a data-driven way. Extensive experimental results on SHREC 2007, SHREC 2011, and SHREC 2014 datasets show that our algorithm achieves superior performance over existing methods.

CCS Concepts: • **Computing methodologies → Shape analysis**.

Additional Key Words and Phrases: 3D point of interest, deep learning, attention mechanism

ACM Reference Format:
Zhenyu Shu, Ling Gao, Shun Yi, Fangyu Wu, Xin Ding, Ting Wan, and Shiqing Xin. 2023. Context-aware 3D Points of Interest Detection via Spatial Attention Mechanism. *ACM Trans. Multimedia Comput. Commun. Appl.* 37, 4, Article 111 (March 2023), 19 pages. https://doi.org/XXXXXXX.XXXXXXX

*Corresponding author.

Authors' addresses: Zhenyu Shu, shuzhenyu@nit.zju.edu.cn, School of Computer and Data Engineering, NingboTech University, Ningbo, Zhejiang, China, 315100 and Ningbo Institute, Zhejiang University, Ningbo, Zhejiang, China, 315100; Ling Gao, 1404919041@qq.com, School of Computer and Data Engineering, NingboTech University, Ningbo, Zhejiang, China, 315100; Shun Yi, ys331_paper@163.com, School of Computer and Data Engineering, NingboTech University, Ningbo, Zhejiang, China, 315100 and School of Mechanical Engineering, Zhejiang University, Hangzhou, Zhejiang, China, 310027; Fangyu Wu, fangyu.wu@zju.edu.cn, School of Computer and Data Engineering, NingboTech University, Ningbo, Zhejiang, China, 315100; Xin Ding, XDing07@163.com, School of Computer and Data Engineering, NingboTech University, Ningbo, Zhejiang, China, 315100; Ting Wan, 771716519@qq.com, School of Computer and Data Engineering, NingboTech University, Ningbo, Zhejiang, China, 315100; Shiqing Xin, xinshiqing@163.com, School of Computer Science and Technology, ShanDong University, Qingdao, Shandong, China, 315100.

1551-6857/2023/3-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

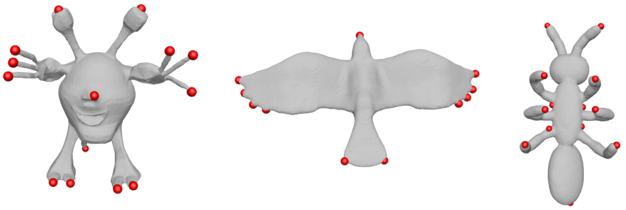

Fig. 1. POIs detected by our method.

1 INTRODUCTION

Extracting points of interest (POIs) is a fundamental problem in multimedia processing. Many tasks rely on robust and effective POI extraction algorithms to obtain satisfactory results, such as viewpoint selection [12], 3D mesh segmentation [20], 3D mesh registration [35, 42], and 3D shape retrieval [4, 6, 39]. Usually, geometric features are regarded as being closely related to judging whether a vertex is a POI on 3D shapes. Therefore, early POIs detection approaches [7, 10] heavily rely on various geometric features of each vertex to effectively extract POIs on 3D shapes.

Recently, data-driven techniques have shown their powerful capabilities in various tasks, such as 3D shape segmentation [29, 31] and 3D shape retrieval [14]. POIs extraction can be deemed as a classification problem, which builds a mapping from the geometric features of each face to whether the face is a POI or not. Therefore, recent POI extraction algorithms begin to focus on employing various data-driven classification approaches, such as decision trees, supported vector machines, and even deep neural networks [30, 32], to achieve satisfactory performance.

Although existing data-driven approaches can obtain better performance than non-data-driven ones, they usually only consider geometric features on a single vertex and lack information from surrounding vertices when learning the relationship between geometric features and POIs.

To analyze and understand complex 3D shapes more effectively, in this paper, we propose a novel POIs detection algorithm by considering both the geometric features of each vertex and the features of its surroundings. To adaptively assign different weights to the features of each vertex and its surroundings in a data-driven way, our algorithm employs the attention context module by introducing the spatial attention mechanism. Extensive experimental results on public datasets show that our method achieves superior performance over existing approaches. Figure 1 shows an example of POIs detected by our method.

The contributions of this paper are three-fold:

- Rather than estimate the POI saliency in a vertex-wise style, in this paper, we also take the spatial context into account, i.e., the POI saliency of a vertex is evaluated with the help of its surrounding vertices, which is more in accordance with human intuition.
- It is difficult to explicitly evaluate how the surrounding vertices influence the base vertex on the POI detection problem. In this paper, we use the attention mechanism to adaptively learn the geometry-aware influence. To our best knowledge, this is the first time that the attention mechanism is used to improve the performance of POI detection.
- Extensive experimental results on multiple datasets show that our algorithm obtains better performance than previous approaches.

The remaining parts of the paper are organized as follows. In Section 2, we briefly review the related work of POIs detection. The details of our method are then described in Section 3. After that, we show extensive experimental results and the comparison between our method and existing

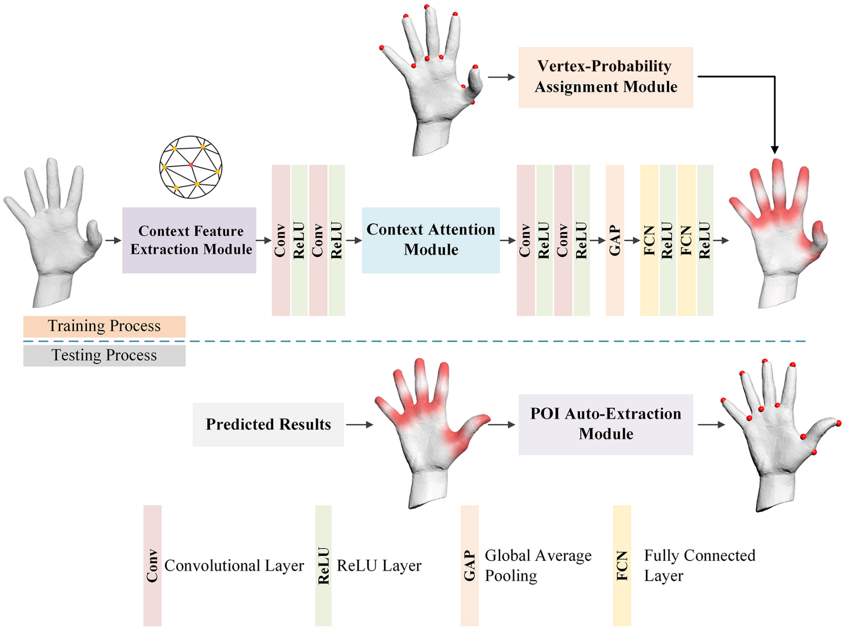

Fig. 2. The overall workflow of our algorithm. Our method takes a 3D model as the input and the predicted probability of each vertex becoming a POI as the output. POIs are extracted by the POI auto-extraction module according to the predicted probability of each vertex.

state-of-the-arts in Section 4. The limitation and future work of our approach is pointed out in Section 5. Finally, we conclude this paper in Section 6.

2 RELATED WORK

POIs extraction originates in computer vision and has also been extensively studied in multimedia processing since then. Generally, POIs have some specific semantic features and are consistent with human visual perception. In recent years, many geometric processing tasks are deemed as relating to POIs, such as shape retrieval [4, 6, 39], facial expression recognition [2], 3D model segmentation [20], and mesh registration [35, 42]. However, since different geometric processing tasks require different POIs annotation results, it is hard to define a common rule to automatically judge whether a given vertex is a POI or not for any purpose. Therefore, it still remains a challenging problem for robust and effective POIs detection.

The geometric features play an important role in distinguishing POIs from other common vertices. Some early work directly detects POIs based on measuring the differences between the geometric features of vertices. Several well-known feature descriptors are often used when extracting POIs on 3D shapes. For example, Zou et al. [44] proposed a shape descriptor based on a Gaussian function, which first computes a Gaussian function on a geodesic-scale space surface, and then obtains the local extrema of the Gaussian function to detect POIs. This descriptor can simultaneously detect POIs on manifold or non-manifold surfaces. Wang et al. [37] also used a Gaussian function to calculate the features of each vertex, and selected all the vertices that could be POIs according to the curvature of the vertex. Furthermore, Lee et al. [22] defined scale-dependent mesh saliency using a center-surround operator on a weighted Gaussian curvature to select salient vertices on the mesh. Gelfand et al. [15] proposed an integral volume descriptor, which integrates the underlying model to obtain eigenvalues, and then extracts POIs on the shape based on this eigenvalue. Inspired by scale-invariant feature transformation, Godila et al. [17] proposed a descriptor of local features of the computational model on a 3D voxel-based model to extract POIs. Castellani et al. [5] also proposed a local feature descriptor, which is defined by a hidden Markov model. The approach assigns the feature to vertices using contextual information, and detects 3D model vertices across multiple views through the similarities of vertices on 3D models.

Recently, with the development of machine learning and deep learning, data-driven approaches are proposed to learn the relationship between the geometric features of vertices and POIs to achieve satisfactory results during extracting POIs. Creusot et al. [9] proposed a method to automatically detect POIs on a 3D face model by using machine learning techniques. The method first uses geometric feature descriptors to calculate feature vectors for every vertex, and then learns a classification model to distinguish between POIs and common vertices, and finally detects POIs on the new shapes by using the learned model. Teran et al. [36] defined the task of POIs detection as a supervised binary classification problem of vertices on a 3D shape. The approach extracts POIs by applying the trained random forest classifier. Nousias et al. [24] presented a novel, efficient saliency detection method for 3D shapes by utilizing baseline 3D importance maps and training corresponding convolutional neural networks. Chen et al. [7] first analyzed the local curvature features and global features of POIs, and then used a random forest algorithm to combine the features into a regression model to predict the location of POIs on new shapes. He et al. [19] proposed a deep Hough voting network to detect 3D POIs on the surface of given 3D shapes. Wei et al. [38] proposed a novel multi-task joint learning network architecture for estimating 3D POIs.

Besides geometric features of vertices, some haptic or projection-based methods are also used to detect POIs on 3D shapes. Lau et al. [21] proposed a tactile-based method for computing vertex saliency, which first maps a 3D model into multiple 2D depth images, and then establishes a regression model between depth images and tactile saliency based on a ranking mechanism, and finally uses the trained network model to predict the protruding vertices on 3D shapes. Shu et al. [32] projected the manually annotated 3D shapes onto 2D images from different angles, then trained a convolutional neural network on these 2D images to predict the probability value of each vertex being a POI or not on 3D shapes, and finally used a clustering method to extract POIs. Although this method can avoid the limitation of geometric feature descriptors, geometric details on 3D shapes are easily occluded during projection. Projecting 3D shapes into multiple 2D views from different angles may relieve the problem in some degree, while increasing the number of views will inevitably lead to a heavier computation burden. Shu et al. [30] designed a deep neural network using stacked auto-encoders, and predicted the probability of each vertex on 3D shapes becoming a POI through the trained network. However, this algorithm only utilizes the geometric features of a single vertex and ignores the feature information of the local context of the vertex.

More recently, the attention mechanism, which can be deemed as a dynamic weight adjustment process, is proposed to imitate the human visual system and has achieved great success in various tasks, such as image segmentation [8], image captioning [41], image dehazing [34], and point cloud recognition [43]. For example, Yu et al. [40] propose a novel multimodal transformer and joint model of self-attention and co-attention interactions for image captioning. Guo et al. [18] present normalized self-attention, which is an effective reparameterization of self-attention and can bring the benefits of the normalization technique inside self-attention. They both achieve superior performance over existing methods on publicly available benchmarks.

In order to fully consider the impact of feature information from surroundings, this paper proposes a novel POIs extraction algorithm based on the spatial attention mechanism, which considers not only the geometric features of vertices themselves, but also the ones from surrounding vertices. We present the context attention module to dynamically assign different weights to the features

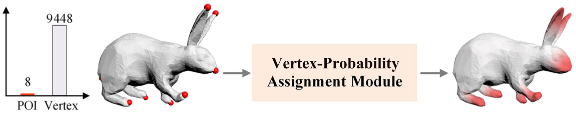

Fig. 3. Generating probability distribution of POIs on a Rabbit model. Note that POIs are far fewer than other normal vertices.

of each vertex and its surroundings in a data-driven way, so that our method can achieve more satisfactory results than existing algorithms.

3 OUR METHOD

3.1 Overview

Figure 2 shows the overall workflow of our method. In the training process, the features of the vertex and its context are used as input, and the label of the vertex is used as the output. In the testing process, the features of the vertex and its context on the test model are used as the input, and the predicted probability of each vertex becoming a POI is the output. We finally obtain POIs on the shape by using the POI auto-extraction module. The computation process of the network can be described as

$$\begin{cases} M = F_{agg}(X), \\ M' = K * M, \\ S_s = F_{att}(M'), \\ p = \phi(W(f_p(K * S_s)) + b). \end{cases} \tag{1}$$

where, X is the feature vector of each vertex, $F_{agg}(\cdot)$ is the context feature extraction module that can aggregate the features of the local context of the vertex into a three-dimensional tensor M, K is the convolution kernel, $*$ is the convolution operation, and $F_{att}(\cdot)$ is the context attention module. S_s is the feature vector after the spatial attention mechanism assigns weights to different features. $f_p(\cdot)$ is a pooling function. W and b are the weights and biases in the fully connected layer. ϕ is the activation function. p is the predicted probability of the vertex. Our method uses the mean squared error loss function to measure the difference between the predicted result and the ground truth, and is optimized by the stochastic gradient descent algorithm. The loss function of our network is

$$Loss = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2, \tag{2}$$

where n represents the number of vertices, $\hat{y}_i$ represents the predicted probability of the vertex, and y_i represents the ground truth of the vertex.

3.2 Generating probability distribution of POIs

Typically, there exist thousands of vertices on a 3D shape. However, only very few vertices are considered as POIs. For example, as shown in Figure 3, a Rabbit model in the SHREC 2011 dataset is only annotated with 8 POIs, while it has 9448 common vertices in total. If we simply regard the problem of POIs extraction as a classification task, it will bring a severe imbalance of training samples in the neural network training process.

To solve the above problem, we assign a probability of being a POI to each vertex on 3D shapes using energy decay through the vertex-probability assignment module. For a vertex p_i , its

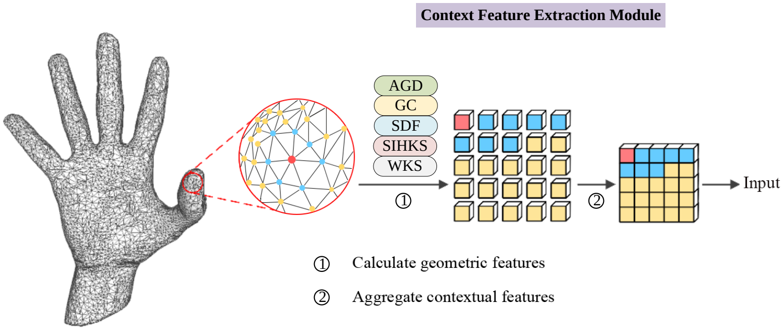
Context Feature Extraction Module
AGD
GC
SDF
SIHKS
WKS
①
②
Input
① Calculate geometric features
② Aggregate contextual features

Fig. 4. Context feature extraction module of POI extraction algorithm. This module encodes the features of the vertex and its surrounding vertices to a matrix and uses them as the input of our deep neural network.

probability ρ_i is calculated as

$$\rho_i = \frac{1}{\sigma\sqrt{2\pi}} \exp(-d^2(v_i, p_i))/(2\sigma^2), \tag{3}$$

where $d(v_i, p_i)$ represents the geodesic distance between the vertex p_i and its closest POI v_i, σ is a parameter that controls the speed of energy decay, and the default value is set to 0.1. The probabilities ρ of all vertices on the surface of the model are normalized. Figure 3 shows an example of generated probabilities according to the manually annotated POIs on a Rabbit model.

3.3 Context Feature Extraction Module

Different from previous learning-based methods [30], our algorithm predicts the probability of a vertex being a POI based on not only the geometric features of a single vertex, but also the features of the vertex's surrounding vertices. For this reason, our algorithm proposes a context feature extraction module to encode the features of vertices and their contexts. As shown in Figure 4, this module mainly includes two steps: extracting vertex features and aggregating context vertices' features. As shown in Figure 4, the feature vectors of each vertex are first extracted using various hand-crafted feature descriptors, including Gaussian Curvature [13] (GC), Average Geodesic Distance [27] (AGD), Shape Diameter Function [28] (SDF), Scale-Invariant Heat Kernel Signature [3] (SIHKS), and Wavelet Kernel Signature [1] (WKS).

Since each vertex's probability of being a POI is related to the geodesic distance between the point and its nearest POI, the context feature extraction module proposed in our algorithm aggregates the contextual feature information of the vertex according to the geodesic distance between vertices. Our algorithm first computes the features of each vertex and then combines them into a feature matrix X of $H \times H \times C$. In X, $X(1,1,*)$ represents the feature vector of the vertex v itself. We then put the features of the neighbor vertices into $X(1,2,*)$, $X(1,3,*)$, $\cdots$, and $X(5,5,*)$ according to geodesic distances.

In this way, the features of each vertex and its neighboring 24 vertices are aggregated in the same feature matrix. The feature matrix is then nonlinearly combined and transformed through convolution operations. Our experimental results show that considering each vertex's context does improve the accuracy of POIs extraction, which is presented in Section 4.

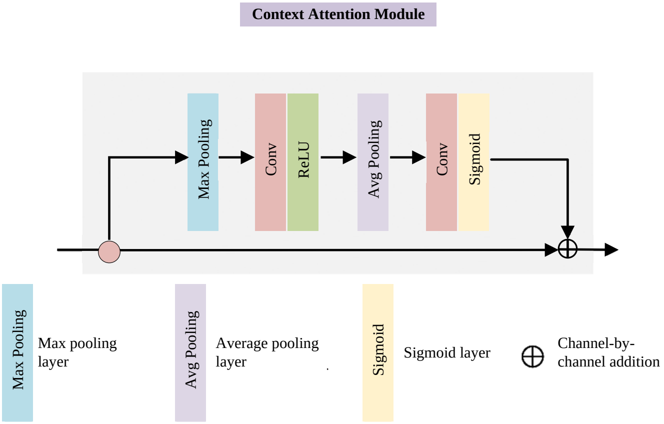

Fig. 5. Context attention module of our POIs extraction algorithm. This module simultaneously considers the feature of each vertex and its local context during computing the probabilities of each vertex by dynamically weights assignment.

3.4 Context Attention Module

During the process of extracting POIs, the POI auto-extraction module selects vertices with local peak values as POIs according to the probabilities of vertices. It means that the probabilities of adjacent vertices will decide whether the vertex can be extracted as a POI. Therefore, in the process of computing the probabilities, we should consider not only the geometric feature of the vertex itself, but also the impact of the local contextual features.

To consider the impact from the context of a vertex, our algorithm proposes a context attention module, which considers the feature of the vertex's context in the process of computing the probabilities of the vertex through the spatial attention mechanism and dynamically weights assignment.

Figure 5 shows the internal structure of the context attention module in detail. In this module, the maximum response of the feature is first calculated by the max pooling layer, and then the non-linear combination of the feature is calculated by the convolution layer and the activation layer. The average value of the features in the receptive field is calculated by the average pooling layer and then passed through the convolution operation. Finally, the convolved features and the original features are added channel by channel. The computation process of this module can be described as

$$\begin{cases} M' = K * F_{max}(M), \\ A = K' * F_{avg}(M'), \\ S_s = F_{add}(M, A), \end{cases} \tag{4}$$

where M is the input feature, $F_{\max}(\cdot)$ is the maximum pooling function which is used to calculate the maximum response of M, K and K' is the convolution kernel, $*$ is the convolution operation, $F_{agv}(\cdot)$ is the average pooling function, and $F_{add}(\cdot)$ is the addition between features of the same size which adds the input features M and A channel by channel. The numbers of filters in each convolution layer are 64, 32, 16, and 8, respectively. The size of each filter is 3*3. The output dimensions of the two FC layers in the network are 4 and 1, respectively. Note that the dimension of features does

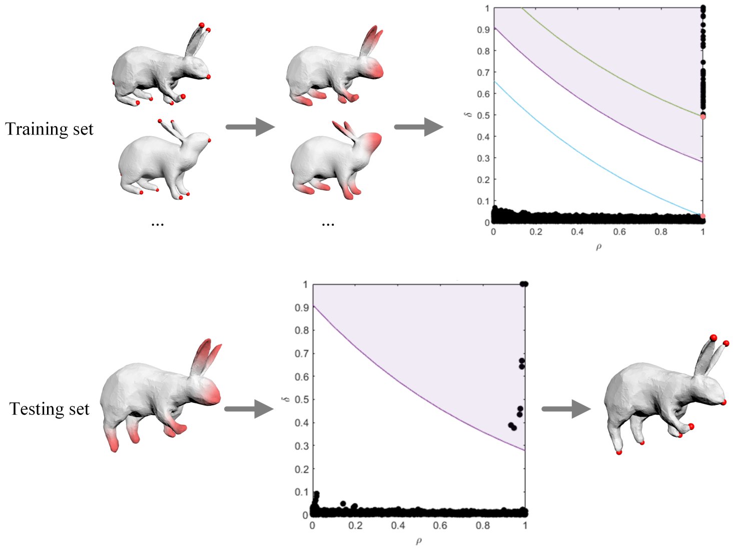

Fig. 6. The curve function used to extract POIs automatically in the decision graph for a Rabbit model.

not change after each layer calculation, which is always 5x5x32 in this paper. This module uses the context attention to simultaneously pay attention to the feature information of the vertex and its local context, and finally obtain the output features S_s.

3.5 POI auto-extraction module

We extract points with a local peak probability on the shape based on the density peak clustering algorithm [26]. The traditional density peak clustering algorithm has two assumptions:

- The density of the density peak points is greater than the density of its neighbor points.
- The distance of different density peak points is relatively far.

The density peak clustering algorithm can find local peak points but lacks the mechanism to automatically extract them.

To realize the automatic extraction of POIs, our algorithm designs an automatic POIs extraction module based on the density peak clustering algorithm. This module takes each vertex's probability of being a POI as the density of the vertices, and the geodesic distance between the vertices as the reference distance. All the vertices are mapped according to the probabilities and distance, into a two-dimensional decision graph, where the horizontal axis of the decision graph represents the probability p of each point. And the vertical axis of the decision graph is the geodesic distance δ from this point to the nearest vertex whose probability is greater than its probability. δ_i is defined as

$$\delta_i = \min_{\rho_j > \rho_i} (d(p_j, p_i)), \tag{5}$$

where $d(p_j, p_i)$ is the geodesic distance between a vertex p_i and the nearest vertex p_j with a probability greater than p_i. Similar to ρ , we also normalize the δ value of each vertex on the same model. It can be seen from Equations 3 and 5 that the larger the ρ value and the δ value of a vertex, the more likely the vertex is a local peak vertex. Accordingly, in the decision graph, the point closer to the upper right corner is more likely to be a POI.

To automatically extract POIs, as shown in Figure 6, our algorithm uses a data-driven strategy to design a curve function during the training phase to automatically separate common vertices and POIs on the decision graph. Since the POIs are distributed in the upper right of the decision graph and the common vertices are distributed in the area close to the horizontal axis, we first use two exponential functions to divide the areas where the common vertices and POIs are located in the training data, and then calculates the offset between the two areas used for automatic extraction. The exponential functions we used are respectively defined as

$$\begin{cases} \delta_1 = e^{-\rho} + b_1, \\ \delta_2 = e^{-\rho} + b_2, \end{cases} \tag{6}$$

where ρ and δ are the horizontal and vertical axes of the decision graph, respectively, b_1 and b_2 are two offsets. As shown in Figure 6, we mark all the vertices in the training set of the Rabbit category on the two-dimensional decision graph, move the curve 1 up, pass through the POI with the smallest δ value, and get the offset b_1. Then, we move the curve 2 down, pass through the common point below, and get the offset b_2.

Finally, the obtained separation curve for automatic POIs extraction is represented as

$$\delta = e^{-\rho} + \frac{b_1 + b_2}{k}, \tag{7}$$

where k is a parameter that adjusts the up and down movement of the separation curve, and its default value is 2. As shown in Figure 6, the points above the separation curve are automatically extracted as the POIs in the testing process.

3.6 Algorithm

Our method can be summarized as Algorithm 1.

Algorithm 1: Context-aware POIs detection Method

```
Inputs:
  3D models in the training set and manually annotated POI.
Outputs:
  The POIs of 3D models in the test set.
Training process:
Step 1: Assign a probability to each vertex of the 3D model with manually annotated POIs
    through the vertex-probability assignment module, and use it as the output of the neural
    network;
Step 2: Compute the geometric features of the vertices on the 3D models through the context
    feature extraction module, and aggregate the context vertices' features as the input of the
    neural network;
Step 3: Train the neural network.
Testing process:
Step 1: Compute the geometric features of the vertices on the 3D model through the context
    feature extraction module to obtain the test input;
Step 2: Use the trained neural network to predict the probability of each vertex;
Step 3: Obtain the POIs on the surface of the 3D model with the POI auto-extraction module.
```

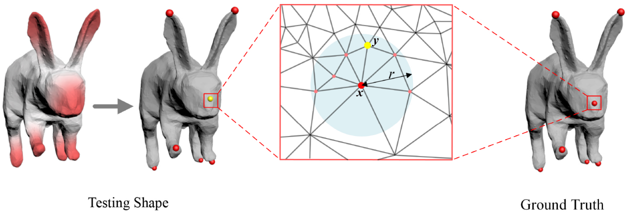

Fig. 7. The POI extracted within the localization error tolerance radius r. It is considered that the POIs extraction is correct if the geodesic distance between the automatically extracted vertex and the ground truth is within r.

4 EVALUATION

4.1 Datasets

We evaluate the performance of our algorithm by testing on the publicly available datasets SHREC 2007 [16], SHREC 2011 [23], and SHREC 2014 [25]. The three datasets all contain non-rigid 3D shapes, which are widely used in various geometry processing algorithms [9, 11, 17]. The SHREC 2007 and SHREC 2011 datasets contain 20 and 30 categories of 3D shapes, respectively, with each category having 20 shapes. The SHREC 2014 dataset contains 400 3D Human models.

4.2 Metrics

To evaluate the performance of our algorithm, we adopt the evaluation method proposed by Dutagaci et al. [10] to measure the performance of the algorithm. This evaluation method has three evaluation metrics, namely false negative error (FNE), false positive error (FPE), and weighted miss error (WME). The three metrics are defined as

$$\begin{aligned} FNE &= 1 - \frac{N_k}{N_g}, \\ FPE &= 1 - \frac{N_k}{N_e}, \\ WME &= 1 - \frac{\sum_{i=1}^{N_g} v_i \cdot m_i}{\sum_{i=1}^{N_g} v_i}, \end{aligned} \tag{8}$$

where v_i represents ground truth POIs, N_g represents the number of ground truth POIs, N_e represents the number of POIs detected by the algorithm, N_k represents the number of vertices within the error tolerance radius r correctly detected by the algorithm. Here, if the geodesic distance between a detected POI and the ground truth is less than or equal to r, then the point is deemed as a correctly detected point. If v_i is correctly extracted, m_i is equal to 1. Otherwise, it is equal to 0. It can be seen from the definitions that the smaller the value of the three evaluation indicators, the better the algorithm performs.

In the evaluation process, the localization error tolerance radius r is an important parameter to represent the geodesic distance range. If the geodesic distance between the point extracted by the algorithm and the ground truth POI is within the range of r, the point is considered correctly extracted. As shown in Figure 7, there exists a manually annotated POI x. Meanwhile, during the detection process, our algorithm selects the adjacent point y, and the geodesic distance between point x and point y is within the range of r. It is considered that the POI extraction is correct.

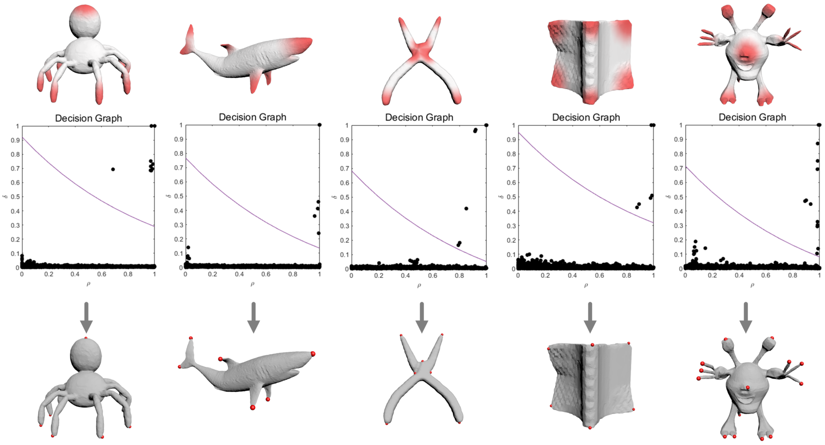

Fig. 8. Extracting POIs from predicted probability distributions on 3D models. The first row demonstrates the predicted probabilities obtained by our neural network. The second row shows the decision graph used to automatically extract POIs. The third row presents the extracted POIs, which are consistent with the manually marked ground truth.

Table 1. Evaluation results of our POI extraction algorithm on SHREC 2011.

Tolerance radius r	Average FNE	Average FPE	Average WME
0.00	0.8929	0.9124	0.9073
0.01	0.5572	0.7350	0.7012
0.02	0.4042	0.4572	0.3442
0.03	0.2721	0.2955	0.2521
0.04	0.1846	0.2081	0.1846
0.05	0.1260	0.1467	0.1260
0.06	0.0903	0.1150	0.0903
0.07	0.0738	0.0988	0.0738
0.08	0.0635	0.0895	0.0635
0.09	0.0550	0.0800	0.0550
0.10	0.0524	0.0772	0.0534
0.11	0.0495	0.0746	0.0525
0.12	0.0455	0.0708	0.0515

4.3 Experimental results

In our experiment, the number of training epochs is 20, the minibatch size is set to 64, and the learning rate is 0.001. Figure 8 shows the predicted results of using the trained neural network on some 3D shapes. It can be seen from the decision graph that our algorithm can clearly distinguish POIs from common vertices. Figure 9 shows the representative results of each category of 3D shapes in the SHREC 2011 dataset. It can be seen that our algorithm can correctly detect the vast majority of POIs.

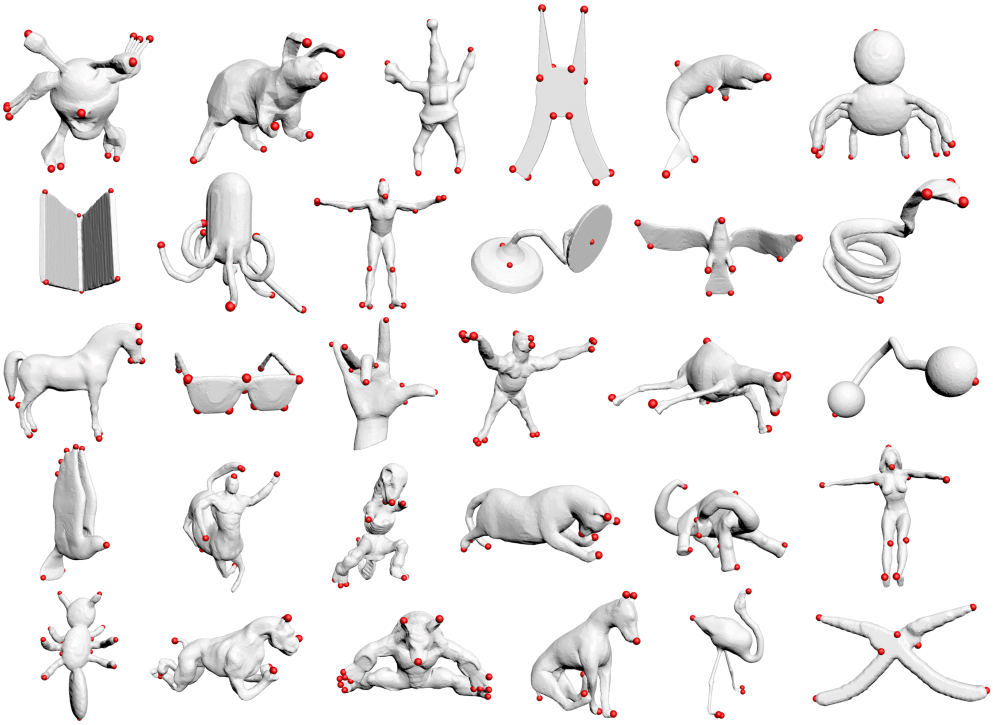

Fig. 9. Representative results of our POIs extraction algorithm on the SHREC 2011 dataset.

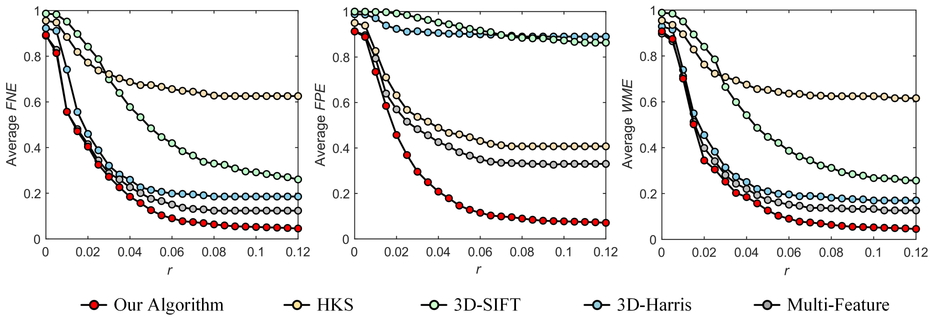

Fig. 10. The comparison of results obtained from our algorithm and existing methods on the SHREC 2011 dataset. The horizontal axis represents the tolerance radius r, and the vertical axis represents FNE, FPE, and WME respectively. Smaller values indicate better performance algorithms achieve.

In addition, we quantitatively measure the performance of our algorithm using the FNE, FPE, and WME metrics. Table 1 presents the measurement results of our algorithm on the SHREC 2011 dataset. It is worth pointing out that each value in the Table represents the average result of the 30 categories of 3D shapes in the SHREC 2011 dataset. Table 1 shows that when the localization error tolerance radius increases, FNE, FPE, and WME decrease rapidly, indicating that our algorithm can correctly extract POIs within a certain error range. When the error tolerance range is greater

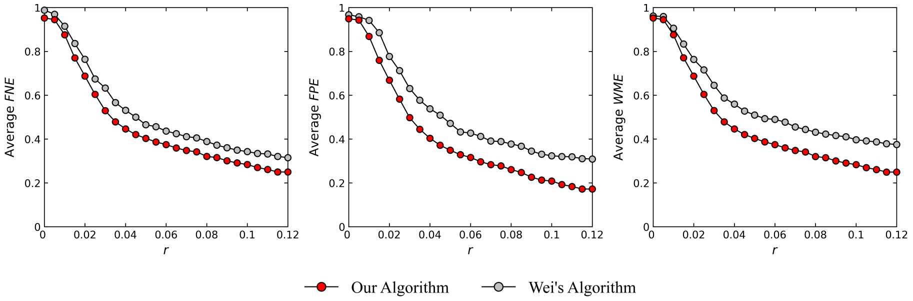

Fig. 11. The comparison of results obtained from our algorithm and Wei's algorithm [38] on the SHREC 2007 dataset. The horizontal axis represents the tolerance radius r, and the vertical axis represents FNE, FPE, and WME respectively. Smaller values indicate better performance algorithms achieve.

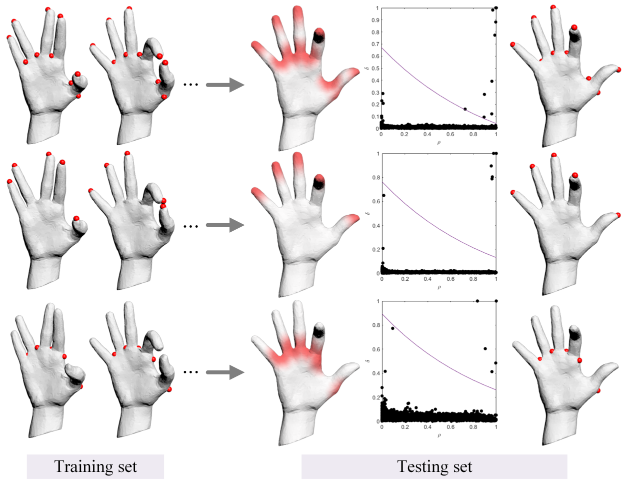

Fig. 12. The extracted POIs results of our algorithm with different training data for Hand models on the SHREC 2011 dataset.

than 0.09, three metrics are stable and close to 0, which indicates that the POIs detected by our algorithm are very close to the human-marked ones within the error tolerance range.

Figure 10 shows the comparison between the results of our algorithm and other POIs detection algorithms, including 3D-SIFT [17], 3D-Harris [33], HKS [10], and multi-feature-based POIs detection algorithm [30]. From the Figure, we can see that the FNE value of 3D-SIFT and 3D-Harris is lower, but the FPE value is higher. It indicates that although these two algorithms can extract most of the POIs, there still exists a lot of common vertices wrongly detected as POIs. In addition, the FPE value of HKS is low, while the FNE value is high, which indicates that the HKS-based algorithm can extract fewer wrong POIs, but miss more correct POIs. The detection results of our algorithm and the multi-feature-based one are significantly better than the other three methods. Besides, our algorithm's values of FNE, FPE, and WME are the lowest, indicating that it obtains the best detection performance. Figure 11 presents the comparison between the results of our approach

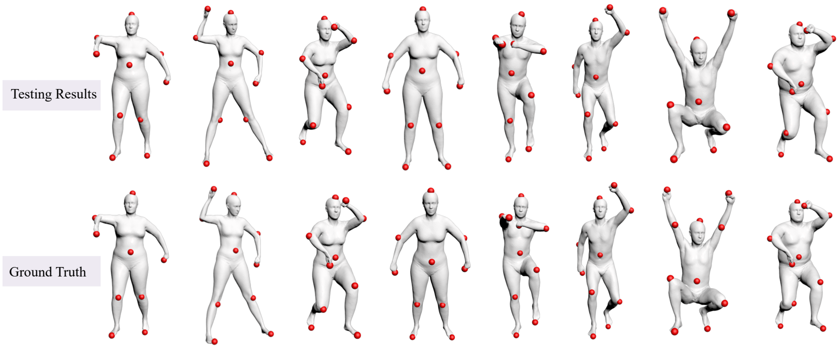

Fig. 13. The comparison of POIs extraction results from our algorithm and the ground truth for Human models on the SHREC 2014 dataset.

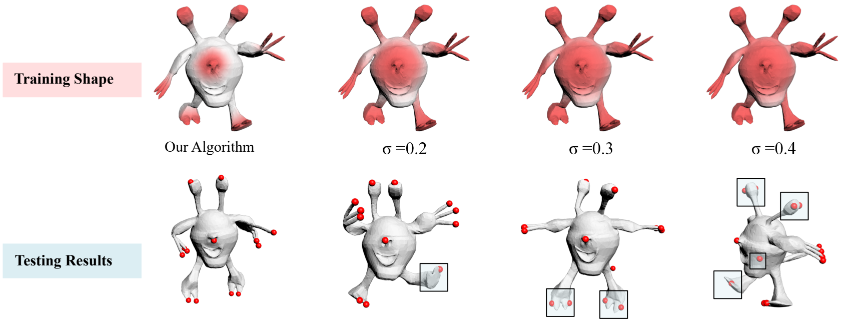

Fig. 14. The impact of the vertex-probability assignment module on the POIs extraction algorithm. The first row shows the effect of different probability distributions for an Alien model in the SHREC 2011 dataset. It can be seen that smaller σ values will make the distribution more concentrated. The second row shows the test results of the algorithm in this section under different σ values.

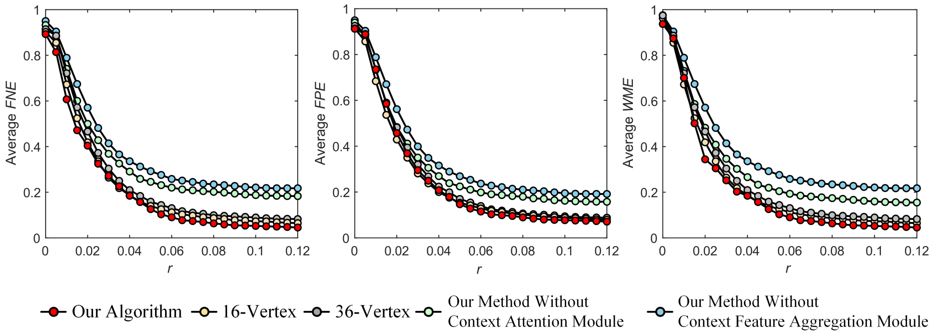

Fig. 15. Ablation experiment of our POIs extraction algorithm on the SHREC 2011 dataset. The horizontal axis represents the tolerance radius r, and the vertical axis represents FNE, FPE, and WME respectively. Smaller values indicate the better performance methods achieve.

Table 2. The average running time of our method on the SHREC 2011 dataset.

Training phase	Feature calculation	Network training	Predicting and extracting
Time (minutes)	8	45	0.5

and a more recent POIs detection method proposed by Wei et al. [38]. From the Figure, one can see that our method also outperforms Wei's algorithm in terms of FNE, FPE, and WME metrics.

Since labeling POIs is a subjective problem, our algorithm adopts a data-driven way to adaptively train the corresponding data and obtain the corresponding POIs extraction results. As shown in Figure 12, whether the fingertips or finger gaps are individually marked as POIs or fully marked as POIs, our algorithm can adaptively extract POIs after training the neural network with corresponding training data. It shows the advantage of our learning-based approach over traditional non-learning-based methods.

To further measure the performance of our algorithm, we perform additional experiments on the SHREC 2014 dataset. Unlike the SHREC 2011 dataset, the SHREC 2014 dataset only contains 400 human models in different poses. We randomly take 25% of shapes (100) as the training set, and the remaining 75% (300) shapes as the test set. The results obtained by our algorithm on the SHREC 2014 dataset are shown in Figure 13. As we can see, our algorithm can obtain satisfactory extraction results on the dataset.

4.4 Performance

We implement our algorithm using MATLAB and C++. And the performance of our approach is measured on a PC with a 2.60 GHz CPU, 32 GB RAM, and an NVIDIA GeForce GTX 2080Ti GPU. Table 2 shows the time cost of the algorithm. On average, it takes about 8 minutes to compute features for 20 shapes, about 45 minutes to train a neural network, and about 0.5 minutes to predict and extract POIs for one shape. It can be seen from the Table that most of the time is spent in the training of neural networks, which accounts for about 85% of the total time.

4.5 Ablation study

We verify and test the effectiveness of the vertex-probability assignment module, the context attention module, and the contextual feature module in our algorithm. Besides, we also investigate the effect of various feature descriptors and different context ranges in our algorithm.

In Section 3, the vertex-probability assignment module uses Equation 3 to assign a probability to each vertex on the model, where the σ value will directly affect the probability distribution on the model surface, which may affect the performance of our algorithm. As shown in Figure 14, the value of σ will affect the distribution of the probabilities on 3D shapes' surfaces and thus affect the algorithm's performance. When σ is 0.1, our algorithm obtains the most satisfactory results. As σ becomes larger, there will be missing extraction and misplacement of POIs. We think the reason is that more vertices assigned larger probabilities will result in more or misplaced local peak points. In general, the value of σ affects the experimental results, but is very limited.

To verify the effectiveness of the context attention module in our algorithm, we use the contextual features of vertices described in Section 3.3 as input, while removing the context attention module described in Section 3.4. Figure 15 shows the extracted results on the SHREC 2011 dataset, where the red curve represents the result of our algorithm, and the turquoise curve is the result of our method without the context attention module. It can be seen from the Figure that the test results

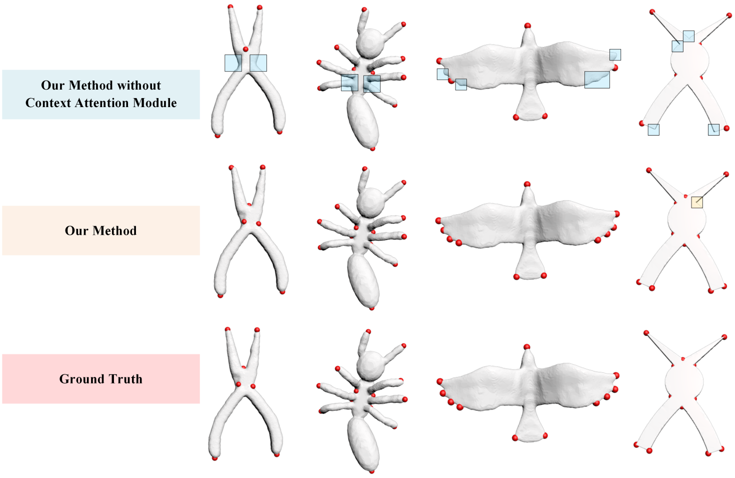

Fig. 16. Impact of spatial attention mechanism on our POIs extraction algorithm. The POIs extraction results on the SHREC 2011 dataset are shown, and the missing POIs are selected with boxes. The first row presents the results of our algorithm without the context attention module, the second row shows the POIs extraction result of our algorithm, and the third row is the human-labeled ground truth.

obtained without the context attention module are higher than our algorithm in terms of FNE and FPE indicators. This shows that the spatial attention mechanism does contribute a lot to improving our algorithm's accuracy.

As shown in Figure 16, if the spatial attention mechanism is removed from the neural network structure, the results of our algorithm will be worse. This is because only vertices with local peak probabilities can be regarded as candidates for POIs. If we do not consider both each vertex and its neighboring vertices simultaneously, some POIs may be missed in the local range. For example, in the joint parts of the Pliers model, the feet of the Ant model, the wing tips of the Bird model, and the joint parts of the Scissors model, there exists more than one POI in these local areas. Without the distinction between the neighborhood feature information getting from the spatial attention mechanism, some important POIs will be missing in the extracted results.

To verify the effectiveness of the context feature extraction module in our algorithm, we remove the context feature extraction module mentioned in Section 3 from our algorithm and re-measure the performance. Note that the spatial attention mechanism is ineffective if no contextual feature is presented. Therefore, we combine the features of each vertex together and represent them as a three-dimensional tensor. The tensors are regarded as the input of the neural network and passed through multiple convolutional layers to predict the probabilities of vertices. As shown in Figure 15, the red curve presents the result of our algorithm, and the blue curve is the result without considering contextual features. We can see that the FNE and FPE increase if not considering the vertex context, which means that the accuracy of POIs extraction results becomes significantly lower.

Figure 17 shows the effect of feature descriptors on randomly selected five categories of 3D shapes, including Flamingo, Glasses, Gorilla, Hand, and Horse, in the SHREC 2011 dataset. We test the performance of our method with four different combinations of feature descriptors, including SDF + SIHKS, SDF + SIHKS + WKS, SDF + SIHKS + WKS + AGD, and SDF + SIHKS + WKS + AGD + GC respectively. We can see that using WKS or AGD can significantly improve the performance of

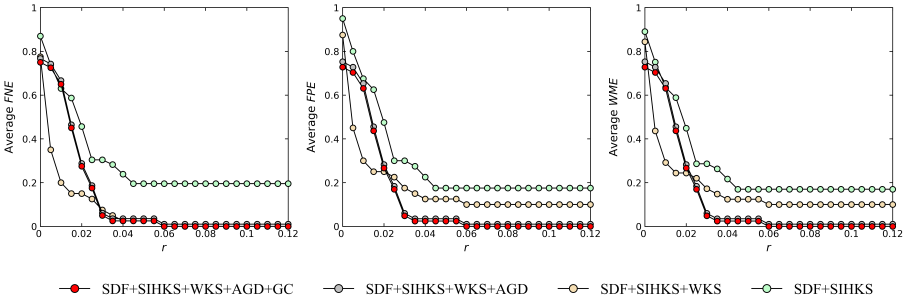

Fig. 17. Ablation experiment about the impact of feature descriptors on randomly selected five categories of 3D shapes, including Flamingo, Glasses, Gorilla, Hand, and Horse, in the SHREC 2011 dataset. The horizontal axis represents the tolerance radius r, and the vertical axis represents FNE, FPE, and WME respectively. Smaller values indicate the better performance methods achieve. The red curve presents the result of our algorithm, and the other curves represent the results of gradually reducing the feature descriptors respectively.

our method. Meanwhile, adding GC can also improve the performance of our method further. Using SDF + SIHKS + WKS + AGD + GC achieves the best performance among various combinations of feature descriptors. Therefore, we finally select SDF + SIHKS + WKS + AGD + GC and use the combination for our method.

To test the effect of different context ranges on the algorithm, we conduct experiments by using 16, 25, and 36 neighboring vertices respectively, according to geodesic distances. In Figure 15, the yellow, red, and gray curves represent the experimental results of our algorithm from using 16, 25, and 36 neighboring vertices respectively. As we can see, using 25 neighborhood vertices achieves the highest accuracy for our algorithm. Therefore, we experimentally selected 25 neighborhood vertices as context ranges for the SHREC 2007, SHREC 2011, and SHREC 2014 datasets.

5 LIMITATION AND FUTURE WORK

Although the algorithm proposed in this paper is effective in various POIs extraction tasks, it does have some limitations.

First, the algorithm in this paper relies on various hand-crafted feature descriptors, which can only be computed on two-dimensional manifolds. Therefore, selecting and designing more effective feature descriptors and extending the algorithm to handle non-manifold 3D models are both our future work.

Second, similar to many other learning-based 3D model shape analysis methods, the neural network in this paper can only be applied to 3D shapes of the same class as the training shapes. Generalizing the algorithm to handle different classes of 3D shapes simultaneously is also our future work.

6 CONCLUSION

This paper proposes a novel POIs extraction algorithm by introducing the spatial attention mechanism. The key to the algorithm lies in the context-aware feature extraction module and the context-aware attention module. By focusing on the local context features of vertices through the spatial attention mechanism, the accuracies of POIs extraction are greatly improved in our algorithm. Extensive experimental results demonstrate that our algorithm is effective on different public datasets and achieves superior performance over previous algorithms.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (62172356, 61872321, 62272277), Zhejiang Provincial Natural Science Foundation of China (LY22F020026), the Ningbo Major Special Projects of the "Science and Technology Innovation 2025" (2020Z005, 2020Z007, 2021Z012, 2019B10128).

REFERENCES

[1] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. 2011. The wave kernel signature: A quantum mechanical approach to shape analysis. In *IEEE International Conference on Computer Vision Workshops*. 1626–1633.

[2] Volker Blanz and Thomas Vetter. 2003. Face recognition based on fitting a 3D morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 9 (2003), 1063–1074.

[3] Michael M Bronstein and Iasonas Kokkinos. 2010. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1704–1711.

[4] Shuhui Bu, Zhenbao Liu, Junwei Han, Jun Wu, and Rongrong Ji. 2014. Learning high-level feature by deep belief networks for 3-D model retrieval and recognition. *IEEE Transactions on Multimedia* 16, 8 (2014), 2154–2167.

[5] Umberto Castellani, Marco Cristani, Simone Fantoni, and Vittorio Murino. 2008. Sparse points matching by combining 3D mesh saliency with statistical descriptors. *Computer Graphics Forum* 27, 2 (2008), 643–652.

[6] Jyun-Yuan Chen, Chao-Hung Lin, Po-Chi Hsu, and Chung-Hao Chen. 2013. Point cloud encoding for 3D building model retrieval. *IEEE Transactions on Multimedia* 16, 2 (2013), 337–345.

[7] Xiaobai Chen, Abulhair Saparov, Bill Pang, and Thomas Funkhouser. 2012. Schelling points on 3D surface meshes. *ACM Transactions on Graphics* 31, 4 (2012), 1–12.

[8] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. 2022. Masked-Attention Mask Transformer for Universal Image Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1290–1299.

[9] Clement Creusot, Nick Pears, and Jim Austin. 2013. A machine-learning approach to keypoint detection and landmarking on 3D meshes. *International Journal of Computer Vision* 102, 1 (2013), 146–179.

[10] Helin Dutagaci, Chun Pan Cheung, and Afzal Godil. 2012. Evaluation of 3D interest point detection techniques via human-generated ground truth. *The Visual Computer* 28, 9 (2012), 901–917.

[11] Helin Dutagaci, Afzal Godil, Petros Daras, Apostolos Axenopoulos, George C Litos, Stavroula Manolopoulou, Keita Goto, Tomohiro Yanagimachi, Yukinori Kurita, Shun Kawamura, et al. 2011. SHREC'11 Track: Generic Shape Retrieval.. In *3DOR@ Eurographics*. 65–69.

[12] Miquel Feixas, Mateu Sbert, and Francisco Gonz A Lez. 2009. A unified information-theoretic framework for viewpoint selection and mesh saliency. *ACM Transactions on Applied Perception* 6, 1 (2009), 1–23.

[13] Ran Gal and Daniel Cohen-Or. 2006. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics* 25, 1 (2006), 130–150.

[14] Zan Gao, Yinming Li, and Shaohua Wan. 2020. Exploring deep learning for view-based 3D model retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications* 16, 1, Article 18 (2020), 21 pages.

[15] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. 2005. Robust Global Registration. In *Symposium on Geometry Processing*. 197–206.

[16] Daniela Giorgi, Silvia Biasotti, and Laura Paraboschi. 2007. Shape retrieval contest 2007: Watertight models track. *SHREC competition* 8, 7 (2007), 7.

[17] Afzal Godil and Asim Imdad Wagan. 2011. Salient local 3D features for 3D shape retrieval. In *Three-Dimensional Imaging, Interaction, and Measurement*, Vol. 7864. SPIE, 275–282.

[18] Longteng Guo, Jing Liu, Xinxin Zhu, Peng Yao, Shichen Lu, and Hanqing Lu. 2020. Normalized and Geometry-Aware Self-Attention Network for Image Captioning. *IEEE* (2020).

[19] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. 2020. PVN3D: A Deep Point-Wise 3D Keypoints Voting Network for 6DoF Pose Estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, WA, USA, 11629–11638.

[20] Sagi Katz, George Leifman, and Ayellet Tal. 2005. Mesh segmentation using feature point and core extraction. *The Visual Computer* 21, 8 (2005), 649–658.

[21] Manfred Lau, Kapil Dev, Weiqi Shi, Julie Dorsey, and Holly Rushmeier. 2016. Tactile mesh saliency. *ACM Transactions on Graphics* 35, 4 (2016), 1–11.

[22] Chang Ha Lee, Amitabh Varshney, and David W. Jacobs. 2005. Mesh saliency. *ACM Transactions on Graphics* 24, 3 (2005), 659–666.

[23] Zhouhui Lian, Afzal Godil, Benjamin Bustos, Mohamed Daoudi, Jeroen Hermans, Shun Kawamura, Yukinori Kurita, Guillaume Lavoué, Hien Van Nguyen, Ryutarou Ohbuchi, et al. 2011. SHREC'11 Track: Shape Retrieval on Non-rigid 3D Watertight Meshes. In *3DOR@ Eurographics*. 79–88.

[24] Stavros Nousias, Gerasimos Arvanitis, Aris Lalos, and Konstantinos Moustakas. 2022. Deep saliency mapping for 3D meshes and applications. *ACM Transactions on Multimedia Computing, Communications, and Applications* (2022).

[25] D. Pickup, X. Sun, P. L. Rosin, R. R. Martin, Z. Cheng, Z. Lian, M. Aono, A. Ben Hamza, A. Bronstein, M. Bronstein, S. Bu, U. Castellani, S. Cheng, V. Garro, A. Giachetti, A. Godil, J. Han, H. Johan, L. Lai, B. Li, C. Li, H. Li, R. Litman, X. Liu, Z. Liu, Y. Lu, A. Tatsuma, and J. Ye. 2014. SHREC'14 track: Shape Retrieval of Non-Rigid 3D Human Models. In *Proceedings of the 7th Eurographics workshop on 3D Object Retrieval (EG 3DOR'14)*. Eurographics Association, 10 pages.

[26] Alex Rodriguez and Alessandro Laio. 2014. Clustering by fast search and find of density peaks. *Science* 344, 6191 (2014), 1492–1496.

[27] Lior Shapira, Shy Shalom, Ariel Shamir, Daniel Cohen-Or, and Hao Zhang. 2010. Contextual part analogies in 3D objects. *International Journal of Computer Vision* 89, 2 (2010), 309–326.

[28] Lior Shapira, Ariel Shamir, and Daniel Cohen-Or. 2008. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer* 24, 4 (2008), 249.

[29] Zhenyu Shu, Xiaoyong Shen, Shiqing Xin, Qingjun Chang, Jieqing Feng, Ladislav Kavan, and Ligang Liu. 2020. Scribble based 3D shape segmentation via weakly-supervised learning. *IEEE Transactions on Visualization and Computer Graphics* 26, 8 (2020), 2671–2682.

[30] Zhenyu Shu, Shiqing Xin, Xin Xu, Ligang Liu, and Ladislav Kavan. 2018. Detecting 3D points of interest using multiple features and stacked auto-encoder. *IEEE Transactions on Visualization and Computer Graphics* 25, 8 (2018), 2583–2596.

[31] Zhenyu Shu, Sipeng Yang, Haoyu Wu, Shiqing Xin, Chaoyi Pang, Ladislav Kavan, and Ligang Liu. 2022. 3D shape segmentation using soft density peak clustering and semi-supervised learning. *Computer-Aided Design* 145 (2022), 103181.

[32] Zhenyu Shu, Sipeng Yang, Shiqing Xin, Chaoyi Pang, Xiaogang Jin, Ladislav Kavan, and Ligang Liu. 2021. Detecting 3D points of interest using projective neural networks. *IEEE Transactions on Multimedia* 24 (2021), 1637–1650.

[33] Ivan Sipiran and Benjamin Bustos. 2010. A Robust 3D Interest Points Detector Based on Harris Operator.. In *3DOR@ Eurographics*. 7–14.

[34] Ziyi Sun, Yunfeng Zhang, Fangxun Bao, Ping Wang, Xunxiang Yao, and Caiming Zhang. 2022. SADnet: Semi-supervised single image dehazing method based on an attention mechanism. *ACM Transactions on Multimedia Computing, Communications, and Applications* 18, 2, Article 58 (2022), 23 pages.

[35] Gary KL Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C Langbein, Yonghuai Liu, David Marshall, Ralph R Martin, Xian-Fang Sun, and Paul L Rosin. 2012. Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics* 19, 7 (2012), 1199–1217.

[36] Leizer Teran and Philippos Mordohai. 2014. 3D interest point detection via discriminative learning. In *European Conference on Computer Vision*. Springer, 159–173.

[37] Chengwei Wang, Dan Kang, Xiuyang Zhao, Lizhi Peng, and Caiming Zhang. 2016. Extraction of feature points on 3D meshes through data gravitation. In *International Conference on Intelligent Computing*. Springer, 601–612.

[38] Guangshun Wei, Long Ma, Chen Wang, Christian Desrosiers, and Yuanfeng Zhou. 2021. Multi-Task Joint Learning of 3D Keypoint Saliency and Correspondence Estimation. *Computer-Aided Design* 141 (2021), 103105.

[39] Jin Xie, Guoxian Dai, and Yi Fang. 2017. Deep multimetric learning for shape-based 3D model retrieval. *IEEE Transactions on Multimedia* 19, 11 (2017), 2463–2474.

[40] J. Yu, J. Li, Z. Yu, and Q. Huang. 2020. Multimodal Transformer With Multi-View Visual Representation for Image Captioning. *IEEE Transactions on Circuits and Systems for Video Technology* 12 (2020).

[41] Jin Yuan, Lei Zhang, Songrui Guo, Yi Xiao, and Zhiyong Li. 2020. Image captioning with a joint attention mechanism by visual concept samples. *ACM Transactions on Multimedia Computing, Communications, and Applications* 16, 3, Article 83 (2020), 22 pages.

[42] Qingkai Zhen, Di Huang, Yunhong Wang, and Liming Chen. 2016. Muscular movement model-based automatic 3D/4D facial expression recognition. *IEEE Transactions on Multimedia* 18, 7 (2016), 1438–1450.

[43] Guanyu Zhu, Yong Zhou, Rui Yao, Hancheng Zhu, and Jiaqi Zhao. 2022. Cyclic self-attention for point cloud recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications* (2022).

[44] Guangyu Zou, Jing Hua, Ming Dong, and Hong Qin. 2008. Surface matching with salient keypoints in geodesic scale space. *Computer Animation and Virtual Worlds* 19, 3-4 (2008), 399–410.

Received 20 November 2022; revised 23 March 2023; accepted 10 April 2023