IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 25, 2019, accepted December 23, 2019, date of publication December 30, 2019,
date of current version January 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962791

Classification of 3D Terracotta Warrior Fragments Based on Deep Learning and Template Guidance

HONGJUAN GAO[1,2] **AND GUOHUA GENG**[1]
[1]School of Information Science and Technology, Northwest University, Xi'an 710127, China
[2]Xinhua College, Ningxia University, Yinchuan 750021, China

Corresponding author: Guohua Geng (ghgeng@nwu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61731015, Grant 61672103, and Grant 61673319, and in part by the National Key Research and Development Projects under Grant 2017YFB.

ABSTRACT The Terracotta Warriors are terracotta sculptures created for China's first emperor more than 2,000 years ago. They are among the most precious unearthed cultural relics of China. However, these relics have been predominantly found in fragments. Fragment classification is currently performed manually on enormous quantities of fragments, which is a time-consuming, inaccurate, and subjective task for archaeologists and conservators. In this study, an automatic method based on a deep learning network combined with template guidance is proposed to classify 3D fragments of the Terracotta Warriors. The fragments are initially classified using PointNet. Then, misclassified fragments are secondly categorized based on their best match to a complete Terracotta Warrior model. Extensive experiments were performed to verify the effectiveness of the proposed method. The promising results demonstrate that the method is the most accurate technique for classifying 3D Terracotta Warrior fragments to date. Moreover, the proposed method can significantly increase the efficiency of future fragment reassembly for the Terracotta Warriors.

INDEX TERMS Data preprocessing, deep learning, 3D fragments classification, intrinsic shape signatures, point cloud, random sample consensus, signature of histograms of orientations, Terracotta warriors.

I. INTRODUCTION

Cultural relics are the embodiment of a national culture and are thus highly valuable for their historical, artistic, and scientific significance. China, an ancient country with a civilization older than 5000 years, has produced a variety of cultural artifacts. For example, the Terracotta Warriors of Qin Shihuang, as shown in Fig. 1, are among the greatest discoveries in the world of archaeological history.

In March 1974, when Chinese farmers were sinking wells for farmland irrigation construction near Xi'an (Shaanxi province), they discovered numerous terracotta fragments. Archaeological excavation showed that the fragments belonged to terracotta figures of warriors and horses dating back to the First Emperor of the Qin dynasty, Shi Huang Di. The figures, facing east and ''ready for battle,'' were individually modeled with their own peculiar characteristics. They were accompanied by their weapons, chariots, and objects of jade and bone. In 1987, the Terracotta Warriors and Horses pit was approved by the United Nations Educational, Scientific and Cultural Organization to be included in the world heritage list and to be known as ''the eighth wonder of the world.''

From 1978 to 1984, the Qin Terracotta Army Archaeological Team of Shaanxi Archaeology Institute excavated Pit No. 1 of these terracotta relics. The excavation area was 2000 square meters, and 1,087 pottery figurines were unearthed. However, after thousands of years of weathering erosion and the collapse of the building, a large number of the excavated relics have been damaged and gathered in piles. As a result, the fragments were more numerous and disordered. At the time of excavation, archaeological technology was unable to restore the unearthed Terracotta Warriors and Horses. In 1985, an archaeological team conducted a second excavation of pit No. 1. Unfortunately, owing to imperfections in the technology and equipment at the time, the excavation lasted only one year. Many of these relics were found in a disintegrated state, requiring many years to restore them, even by experienced archaeologists. This endeavor was especially challenging since some pieces were missing. Consequently, in the first two excavations, only a small number of fragments were manually restored. More recently, with the rapid

The associate editor coordinating the review of this manuscript and approving it for publication was Shadi Alawneh.

FIGURE 1. Terracotta Warriors.

development of three-dimensional (3D) electronic and computer visualization technologies, the restoration of cultural relics has no longer entirely relied on manual methods. In 2009, Pit No. 1 was excavated for the third time. At this point, the Visualization Institute of Northwest University, Shaanxi Province, China worked with Emperor Qinshihuang's Mausoleum Site Museum to restore the damaged Terracotta Warrior fragments by computer technology.

Computer-aided restoration of damaged Terracotta Warriors is divided into two major tasks. The first task is the classification of the fragments into different groups according to their body part. The second is the reconstruction of those fragments into the original archaeological objects. In this study, we considered the first task. Fragment classification is a crucial step in the restoration process. The classification accuracy of fragments directly affects the precision and efficiency of the subsequent fragment reassembly.

The remainder of this paper is organized as follows. In Section 2, existing methods for classification of archaeological fragments are examined. In Section 3, data processing methods are described. In Section 4, the use of PointNet architecture is introduced. In Section 5, details of the proposed method are provided. In Section 6, experimental results and analysis are presented. Finally, conclusions are given in Section 7.

II. RELATED WORK

A. CLASSIFICATION OF ARCHAEOLOGICAL FRAGMENTS

In the restoration of cultural relics, fragment classification is a precondition of fragment reassembly. To date, various methods have been proposed to reassemble archaeological sherds. Nevertheless, few researchers have strived to classify archeological fragments.

Researchers have achieved the classification of fragments by taking into account their shape, color, texture, decoration, technological elements, and material characteristics. Kampel and Sablatnik [1] classified archaeological sherds by estimating their color. This method requires color calibration with known illuminants. Therefore, color estimation is very sensitive to lighting variations. Kampel *et al.* [2] proposed the use of two-dimensional (2D) profiling and segmenting the full profile into relevant subparts to classify archeological pottery sherds. Smith *et al.* [3] approached the problem on the basis of color and texture characteristics. Color similarity and texture similarity between sherd images are determined by estimating a color histogram and applying geometric total variation energy (TVG). Then, a sherd image descriptor vector is generated as a combination of TVG and color histograms. The proposed descriptor accurately represents texture. Nevertheless, it achieved poor classification rates for images with low amounts of texture and minor color and intensity variations. Qi and Wang [4] applied Gabor wavelet transformations to extract image features based on texture. They then classified the sherds using a fuzzy C-means algorithm. Karasik and Smilansky [5] focused on the classification of ceramic assemblages on the basis of their profile morphology. The technique proposed herein is based on the assumption that the shapes of sherds can be entirely characterized by their profiles. Similarly, Zhou *et al.* [6] focused on the color features of porcelain image and developed a system for Yao Zhou's porcelain classification. Makridis and Daras [7] extracted color information and local texture features, while employing a bag-of-words technique to construct a global vector representing the whole sherd image. Their technique exploits both front and back views of the sherds to increase their classification accuracy. Oxholm and Nishino [8] reassembled thin artifacts through the photometric properties of the boundary contours. Rasheed and Nordin [9] used intersections of RGB colors among archeological fragments to extract the fragment texture features. They obtained a high level of accuracy by classifying Euclidean distances between the texture and color features. Sablatnig *et al.* [10] proposed a bottom-up method to classify fragments using a description language comprised of primitives (with certain properties, such as length) and relations among these primitives (such as the curvature of connecting points and positions). Kang *et al.* [11] proposed a classification method based on salient geometric features, which they matched by utilizing an empirical mode decomposition (EMD) method. The experimental results demonstrated that their method is highly accurate for the classification of Terracotta Warrior fragments. Yang *et al.* [12] extracted texture features of the fragment images by using a public scale-invariant feature transform (SIFT) algorithm. They classified the fragments using a support vector machine. The experimental results indicated that this method can significantly improve the accuracy of fragment classification. Na [13] proposed a semi-supervised classification algorithm of a manifold regularization multi-core model to classify fragment images. This method enabled the subsequent splicing and restoration of the Terracotta Warriors. Wang [14] proposed placing 2D images of the Terracotta Warriors into a convolutional neural network for data training. This approach avoids artificial feature extraction. Liu [15] proposed a 3D residual neural network algorithm for determining the parts of the Terracotta Warriors to which the sherds belong. In the fragment identification task, their experimental results

showed that the recognition accuracy rate of their 3D residual neural network method was 83.59%, thereby fulfilling the requirement for fragment identification.

B. DEEP LEARNING METHODS FOR 3D DATA

Traditional convolutional neural networks (CNNs) require samples to appear at fixed spatial orientations and distances in order to facilitate the convolution. From a data structure point of view, a point cloud is an irregular and unordered set of vectors. Thus, directly convolving kernels against features associated with the points will result in the loss of shape information and variance to point ordering.

Most work in deep learning focuses on regular input representations, such as voxels, images, and meshes. Various authors [16]–[18] have applied 3D CNNs to voxelized shapes. However, volumetric representation is limited by its resolution on account of the computational complexity of 3D convolution and data sparsity. Multiview CNNs, proposed in [19], apply 2D CNNs to classify a collection of 2D images that from 3D shapes rendered views, and have achieved good performance on shape classification and retrieval tasks. A couple recent studies [20]–[21] used spectral CNNs on meshes. However, these approaches are currently constrained on manifold meshes, such as organic objects. In another study [22], 3D data were converted into a vector, extracting concise but geometrically informative shape descriptors to guide the deep neural network training. Nonetheless, this method may be constrained by the representative power of the features extracted.

In the above applications, point cloud data are often converted to voxel or mesh representations before they are fed into the deep network. However, this data representation transformation renders the resulting data unnecessarily voluminous and introduces quantization artifacts that can obscure natural data invariances. If directly dealing with point clouds, it is easy to apply transformations, such as translation and rotation, as differentiable layers to achieve continuity invariance. The study in [23] examined this problem. A deep network architecture called PointNet, which directly consumes point clouds, was designed. The key to this approach is to achieve input order invariance by using a symmetric function on transformed elements in the set. Although simple, PointNet is highly effective and efficient. PointNet++ [24] applies PointNet hierarchically for better capturing of local structures. Another recent method [25] learns an χ-transformation from the input data and then applies it to simultaneously permute and weigh the input features. The transformed features are subsequently applied to the convolution neural network.

III. CONTRIBUTIONS

Building on the above advancements, this paper presents a classification approach of 3D archaeological fragments. The contributions of this work are as follows:

To the best of our knowledge, this is the first work to apply PointNet to classify 3D fragments in the field of digital archaeology. The method trains an end-to-end deep learning networks to automatically extract and describe features of Terracotta Warrior fragments. Although some expertise is required to tune the optimization hyper-parameters, no human intervention is required for feature extraction. Both feature extraction and classification are trained automatically.

For each fragment misclassified by PointNet, a second classification is implemented to determine the part to which the fragment belongs by comparing the prospectively complete Terracotta Warrior with the fragment. This step further improves fragment classification accuracy.

The proposed approach can be applied to the task of classifying 3D archaeological fragments and especially can effectively distinguish fragments with very similar shapes. The proposed method marks a solution to an extremely difficult and time-consuming task for conservators and archaeologists.

IV. PRELIMINARIES

The point cloud is a widely used format of 3D model representation. We acquired 3D data of the Terracotta Warriors by using an Artec3D Scanner from Emperor Qinshihuang's Mausoleum Site Museum, as shown in Fig. 2.

FIGURE 2. Artec3D scanner.

A. ORIGINAL DATA PRETREATMENT

In the process of scanning the target objects with the Artec3D scanner, defects, such as noise points, speckles, and holes appeared, as displayed in Fig. 3(a) and (b).

Owing to the distance between the scanner and target object, sight interference and occlusion of other objects generated noise points in the scanned model. This noise directly affected the quality of the data.

The surfaces of some Terracotta Warrior fragments are not smooth (they have minimal roughness). This may cause the phenomenon of reflection or diffuse reflection when the laser irradiates the surface of the target object in the scanning process, resulting in the formation of holes. Moreover, errors caused by the scanner itself, such as jitter, preheating, and other issues, also lead to the formation of holes.

The occlusion on some of the surfaces of the Terracotta Warrior fragments, coupled with the fact that the scanner could not collect data from any direction, resulted in incomplete scanning. To obtain complete scans, the fragments had to be scanned from multiple angles. Once this was

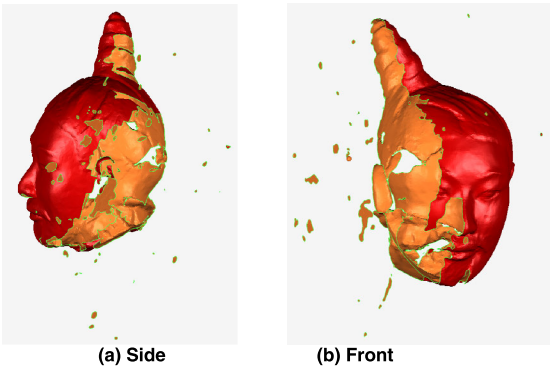


FIGURE 3. Defects in the raw data from the 3D scans.

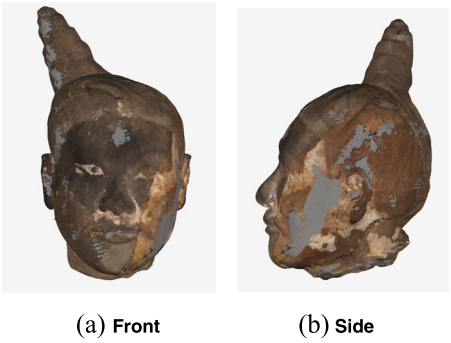


FIGURE 4. Result of data pretreatment.

accomplished, the scanned data from multiple angles were aligned and registered.

This data pretreatment was necessary to improve the data quality. The specific technical preprocessing steps we employed using Geomagic software are listed below:

(1) The raw scanned data were aligned and registered;
(2) Isolated points outside of the cloud were deleted in vitro;
(3) Noise and speckles were filtered;
(4) Holes in the 3D model surfaces were patched.

Fig. 4 (a) and (b) show the effect of data preprocessing on the head fragments of one of the Terracotta Warriors.

B. DATA SAMPLING

A point cloud consists of large numbers of data points with a dense distribution; it thus cannot be directly applied to the framework of neural networks. Sampling and normalization of the 3D point cloud provide standardized data for the subsequent use of the neural networks. Fig. 5 displays the result of data sampling on the arm fragments of one of the Terracotta Warriors.

V. POINTNET ARCHITECTURE COMPONENTS

The success of employing deep neural networks for image processing has motivated a data-driven approach to learning features of point clouds. Deep point cloud processing and analysis methods have rapidly developed and have outperformed traditional approaches in various tasks.

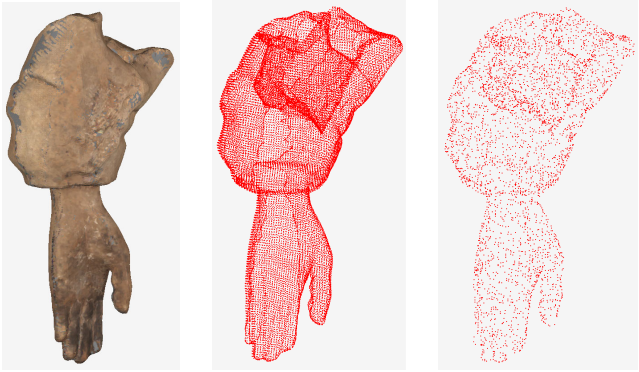


FIGURE 5. Result of data sampling.

Our work is related to recent advances in PointNet [23]. PointNet, which directly feeds the point cloud data, can approximate any set function that is continuous and directly map the input to the target classification by passing the input through multiple layers. It can also summarize an input point cloud through a sparse set of key points, which corresponds to the skeleton of the visualization of an object. PointNet is highly robust to small perturbations of input points and to corruption through point insertion or deletion.

The basic PointNet architecture is simple, as all the stages are automatically processed. Fig. 6 shows a flowchart of PointNet, which takes points from the point cloud as input, applies input and feature transformations, and then aggregates point features by max pooling. The initial layers are designed to extract the useful level features (low to high), and the succeeding layers map the extracted features to the target classification. The maximum pool layer is a symmetric function that aggregates information from all points.

A. CONVOLUTIONAL LAYER

Convolutional layers, the main building blocks of CNNs, learn the feature representation of the input points by performing convolutions over the input.

The extracted feature maps are computed by convolving the input data with the kernels and adding the bias parameters to the features. In mathematical terms, consider x as the input data, w as the kernel, and b as the bias for the convolutional layer. Feature map z generated from this layer is calculated as:

$$z = wx + b. \tag{1}$$

The point cloud structure is completely different from the image. A point cloud is represented as a set of 3D points $A = \{p_i | i = 1, 2, \ldots, n\}$, where each point p_i is a vector of its (x, y, z) coordinate plus extra feature channels, such as its normal and color.

The convolutional layer consists of several kernels that are applied to calculate different features from the input data. For simplicity and clarity, we only use x, y, z coordinates as the feature channels of the given point. Therefore, there are only

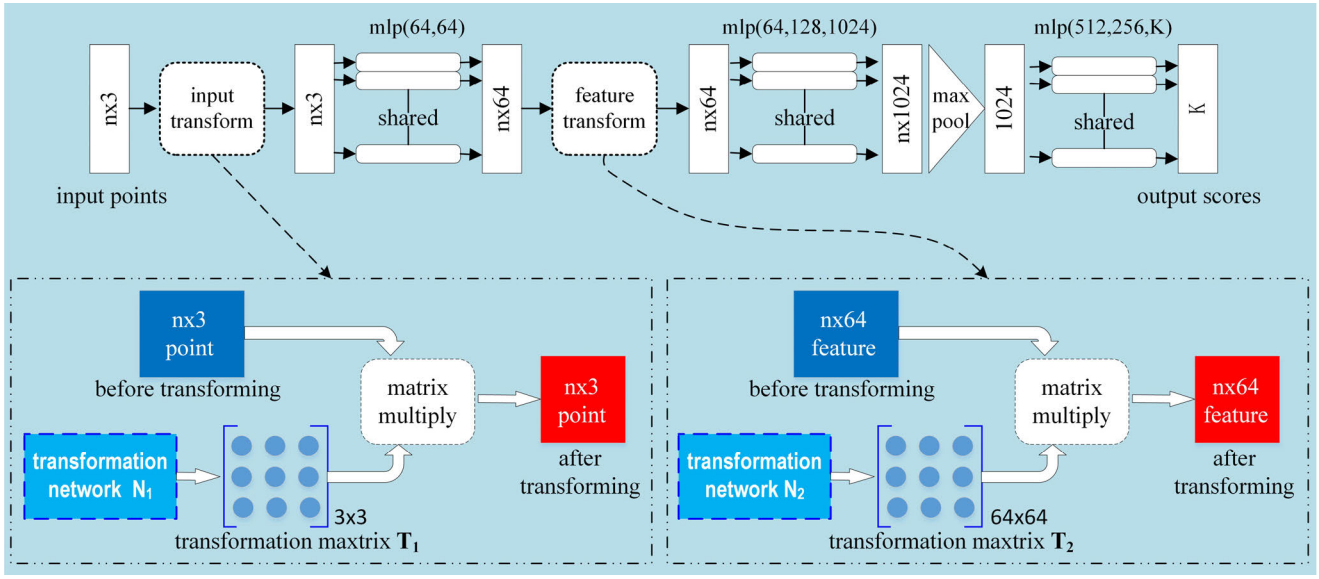


FIGURE 6. PointNet architecture. The numbers in parentheses are layer sizes (mlp: multi-layer perceptron).

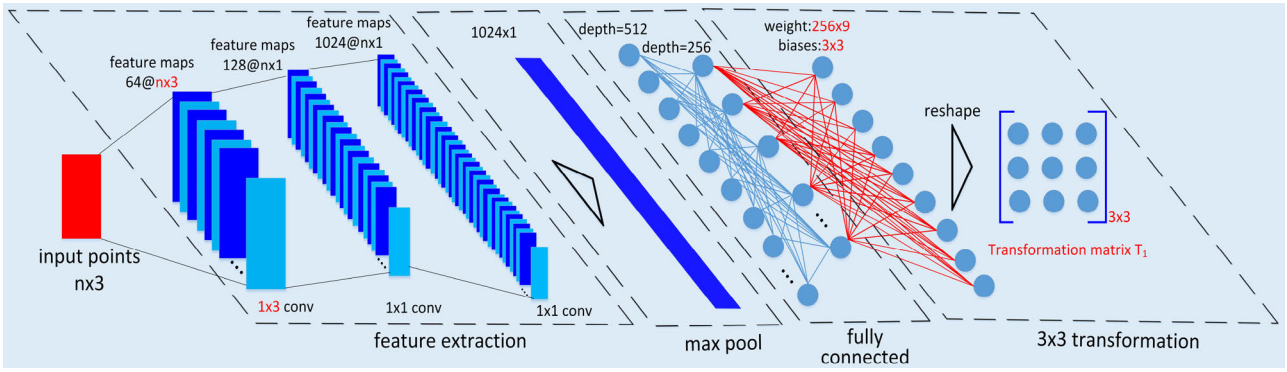


FIGURE 7. Affine transformation matrix T_1 as predicted by a mini-network.

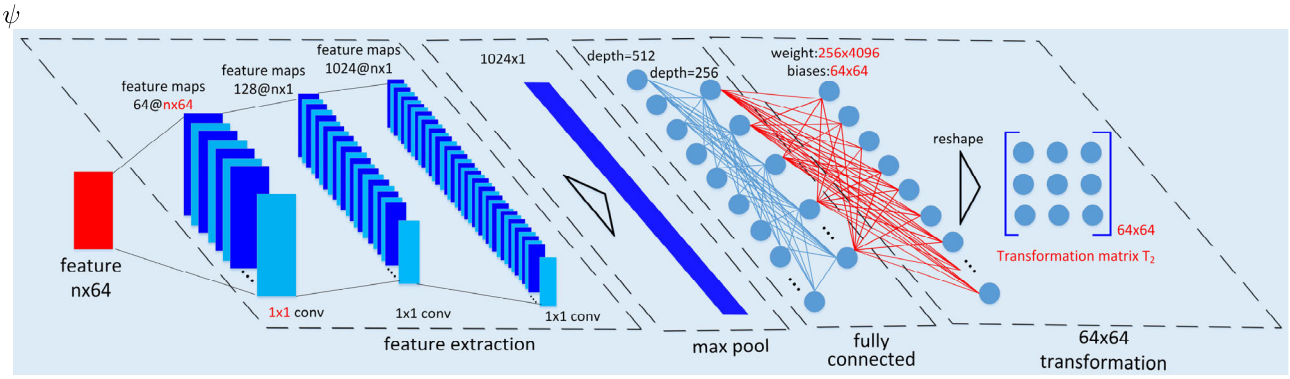


FIGURE 8. Affine transformation matrix T_2 as predicted by a mini-network.

two kinds of convolution cores (1×3 and 1×1) in PointNet (see Fig. 7 and Fig. 8).

B. ACTIVATION FUNCTION

The key of every deep network is a linear transformation followed by an activation function. The activation functions follow the convolutional layers to detect the nonlinearities in the deep network. The activation function plays an important role in improving the performance of the training task. Rectified Linear Unit (ReLU) is the most successful and popular activation function used in neural networks [26]–[28]. It is defined as

$$f(x) = \max(x, 0), \tag{2}$$

where $f(x)$ is the input to the activation function and x is the output. ReLU retains the positive value of the input and changes the negative value to zero.

Owing to its effectiveness and simplicity, ReLU has been the default activation function used by the deep learning community. Deep networks with ReLUs are more easily optimized than networks with sigmoid or tanh units. Therefore, for all layers except the last one, ReLUs are used in PointNet to quickly train deep neural networks, which alleviates the difficulties of weight-initialization and vanishing gradients.

C. MAX POOLING

The use of a single symmetric function, max pooling, is a key aspect of PointNet. The network learns a set of optimization functions that select interesting or informative points in the point cloud, and it then encodes the reason for their selection.

In contrast to pixel arrays in images or voxel arrays in volumetric grids, a point cloud is a set of unordered points that are not isolated. Thus, local structures from nearby points contain meaningful information. As a geometric object, the learned representation of the point set should be invariant to rotating or translating transformations. Therefore, the model must be able to approximate a general function defined on a point set by applying a symmetric function on transformed elements in the set. Hence,

$$f(\{x_1, x_2, \ldots, x_n\}) \approx g(h(x_1), \ldots, h(x_n)), \tag{3}$$

where $f : 2^{\mathbb{R}^N} \to \mathbb{R}, h : \mathbb{R}^N \to \mathbb{R}^K$ and $g{:}\mathbb{R}^K \times \cdots \times \mathbb{R}^K \to \mathbb{R}$ are symmetric functions. Function h is approximated by a multi-layer perceptron network, while g is approximated by the composition of a single variable function and a max pooling function. Through a collection of h, different properties of the set are captured by a number of functions f.

D. FULLY CONNECTED LAYERS

The final fully connected layers of the network aggregate the learned optimal values into the global descriptor for the entire shape as the shape classification. Dropout layers are used for the last multi-layer perceptron network in the classification net.

E. TRANSFORMATION NETWORK

The basic architecture of PointNet was detailed in the previous section. In this section, details on the transformation network and training parameters are the focus.

Since each point transformation is independent, it is easy to apply rigid or affine transformations to the input format of the point clouds. Thus, to further improve the results, PointNet adds a data-dependent spatial transformer network that attempts to canonicalize the data before the multilayer perceptron processes them. Affine transformation matrix T_1 is planned by a mini-network. PointNet directly implements this transformation to the coordinates of the input points. The mini-network itself resembles a vast network and is composed of basic modules of point-independent feature extractions, max pooling, and fully connected layers. The first transformation network is a mini-net that takes the raw point cloud as input and regresses it to a 3×3 matrix. It consists of a shared multi-layer perceptron network (64,128,1024) on each point with layer output sizes of 64, 128, and 1024; a max pooling across points; and two fully connected layers with output sizes of 512 and 256. Here, the output matrix is initialized as an identity matrix. More details about the mini-network are shown in Fig. 7. The above concept has yet to be extended to the alignment of feature space as well. PointNet has another alignment network on point features that predicts feature transformation matrix T_2 to arrange features from different input point clouds. The feature transformation network has the same architecture as the input transformation network, except that the output is a 64×64 matrix. The matrix is further initialized as an identity. Details about the alignment network are shown in Fig. 8.

However, the dimension of the transformation matrix in the feature space is higher than the spatial transformation matrix, which makes optimization difficult. Thus, a regularization term is added to the softmax training loss, which constrains the feature transformation matrix to be close to an orthogonal matrix:

$$L_{reg} = \left\| I - AA^T \right\|_F^2. \tag{4}$$

Here, A is feature alignment matrix T_2. It is ideal for an orthogonal transformation to maintain the information in the input. By adding the regularization term, the model achieves better performance and the optimization becomes more stable. A regularization loss with a weight of 0.001 is added to the classification loss of softmax to make the matrix close to orthogonal.

VI. PROPOSED METHOD

The proposed method incorporates PointNet and the template-guide approach [29]. This method is formulated in a three-step pipeline as follows. First, to solve the problem of insufficient sample sizes of fragments, data enhancement is performed by Monte Carlo sampling. Second, fragments of the Terracotta Warriors are labeled depending on their body parts, and the point cloud data of the fragments are directly fed into the PointNet training model and classified. Third, the misclassified fragments are classified for a second time by determining the corresponding relationship between the Terracotta Warrior ''templates'' and the fragments.

The classification performance of PointNet has been proven on public dataset ModelNet40 [30]. On account of the uniqueness of the Terracotta Warrior data, it is not comparable to the ModelNet40 dataset. As a result, PointNet classification of the Terracotta Warrior dataset has inferior quality. For example, as shown in Fig. 9, some fragments are very similar in shape; nevertheless, PointNet assigns them to different parts of the terracotta structures, resulting in misclassification.

Several of the Terracotta Warriors have similar geometric shapes and surface textures. The relatively well-preserved

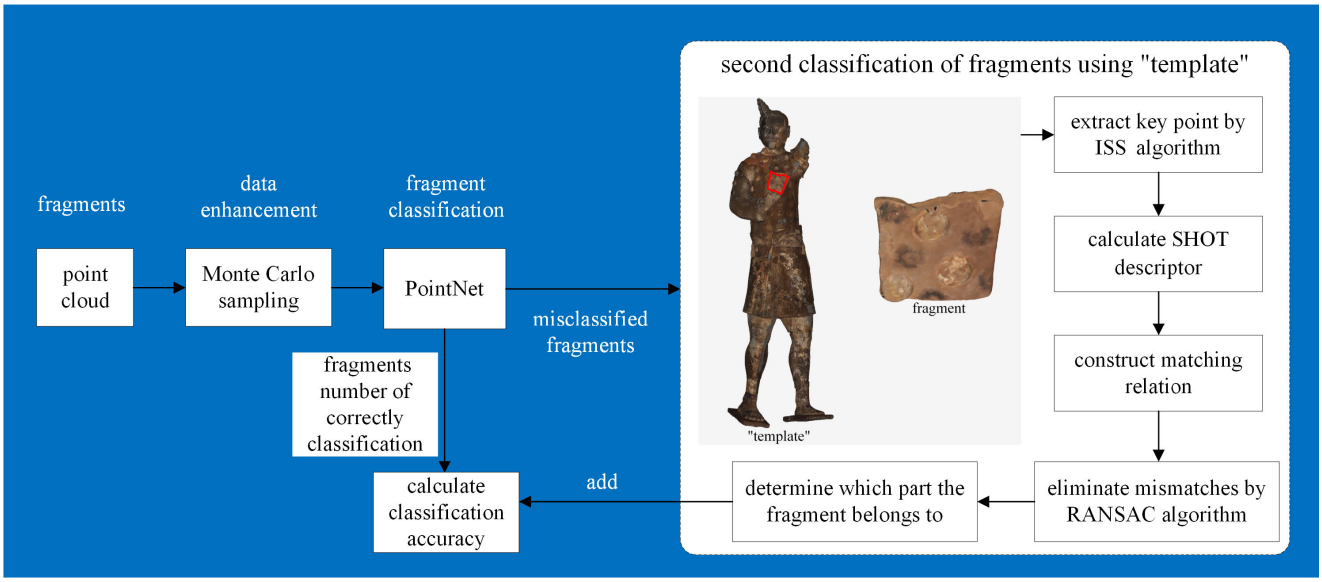


FIGURE 9. Framework of the proposed method.

Terracotta Warriors are deemed "templates." By calculating the relationship between the template and the fragment, effective guidance for the classification of 3D fragments can be provided, which further improves the classification accuracy.

The second classification of the Terracotta Warrior fragments that were misclassified is performed as follows. First, the Intrinsic Shape Signatures (ISS) algorithm is used to extract key points in the intact regions of the Terracotta Warrior fragments and template. Then the SHOT feature descriptors of key points are calculated. Next, the corresponding positions of the fragments on the template are determined by comparing the similarities of the SHOT feature descriptors.

A. DATA ENHANCEMENT

Most deep learning models require a large number of training samples. However, in this case, there is an insufficient number of fragment samples. Therefore, Monte Carlo sampling is employed to augment the training data and to match them to the deep learning network structure.

Monte Carlo sampling is a random sampling method based on the Monte Carlo method. It is a fundamental method for 3D point cloud data processing. The Monte Carlo method approximates an objective function or estimates a certain statistical quantity that may otherwise be difficult to obtain by simulating random variables from a specific statistical model and probability distribution.

The point cloud model can be represented as different point cloud subsets by Monte Carlo sampling. Point cloud model P can be regarded as point set $\mathrm{P} = \{P_i | i = 1, 2, \ldots, n\}$, where the coordinates of each point are (x, y, z). The m points are randomly selected from the point cloud model as a subset $\mathrm{A} = \{A_j | j = 1, 2, \ldots, m\}$ by using the Monte Carlo sampling method. In the same way, subsets $\mathrm{B} = \{B_j | j = 1, 2, \ldots, m\}$, $\mathrm{C} = \{C_j | = 1, 2, \ldots, m\}$, $\mathrm{D} = \{D_j | = 1, 2, \ldots, m\}$, and so on, can be obtained. These subsets represent the same point cloud model; however, they are not equal on account of the random sampling. Since PointNet directly feeds point clouds, data enhancement can be accomplished by transforming one point cloud model into multiple subsets through Monte Carlo sampling.

B. EXTRACTING KEY POINTS BY ISS ALGORITHM

The key points on the point cloud are those with stability and saliency. In the proposed approach, the ISS algorithm [31] is used to extract feature points on fragment F_i and template M. The purpose of extracting key points by the ISS algorithm is three-fold: to construct its covariance matrix for each point in a certain neighborhood of the support region of each 3D point, to obtain the eigenvalues and eigenvectors through covariance analysis, and to calculate the ratios of the largest eigenvalue to the second largest eigenvalue and the second largest eigenvalue to the smallest eigenvalue. If both of those ratios are less than a certain threshold, then the point is selected as the key point of the 3D fragment surface. Fig. 13 illustrates an example of extracted features.

To eliminate the influence of the non-uniform density of the 3D point cloud, a weight is defined. The weight of points in a sparse area is larger than that of points in a dense area. If a sphere with radius R is defined for each point P_i, the weight is the reciprocal of the number of points in Pi's sphere:

$$w_i = 1 \Big/ \left\| \left\{ p_j : \left| p_j - p_i \right| < R \right\} \right\|. \tag{5}$$

Next, the covariance matrix COV (P_i) of each point P_i and all points P_j within the radius R are calculated:

$$\mathrm{cov}\left(p_i\right) = \sum_{\left|p_j - p_i\right| < R} w_j \left(p_j - p_i\right) \left(p_j - p_i\right)^T \Big/ \sum_{\left|p_j - p_i\right| < R} w_j. \tag{6}$$

The eigenvalues of COV (P_i) are then calculated and arranged in descending order, and the thresholds γ_{21} and γ_{32} are set. If point P_i satisfies both Equation (7) and Equation (8), P_i is selected as the key point:

$$\lambda_i^2 / \lambda_i^1 \leq \gamma_{21}, \tag{7}$$

$$\lambda_i^3 / \lambda_i^2 \leq \gamma_{32}. \tag{8}$$

C. COMPUTATION OF SHOT DESCRIPTOR

To accommodate geometric differences between the fragments and templates, a 3D shape descriptor that is insensitive to local geometric variance is preferred. A histogram-based descriptor, such as shape context [32], spin images [33], and Signature of Histograms of Orientations (SHOT) [34], has been the preferred option for this purpose. SHOT can stably reflect geometric variance and outperforms other local descriptors in terms of shape retrieval, object recognition, and 3D reconstruction.

Construction of a SHOT descriptor is divided into two procedures: feature coding (signature) and histogram statistics (histogram). Feature coding is the key. The point cloud feature descriptor encodes the geometric information (including the normal direction, angle, and curvature of the K-nearest point), color, and texture information. The histogram is used to describe the distribution of features and enhance the robustness of features (from the perspective of probability).

The definition of a Local Reference Frame (LRF), invariant to translations and rotations and robust to noise and clutter, has been the preferred option to endow a 3D descriptor with invariance to the same sources of variations, similarly to the way rotation and/or scale invariance is injected into 2D descriptors.

1) LOCAL REFERENCE FRAME FROM DISAMBIGUATED EIGENVALUE DECOMPOSITION

SHOT constructs a unique and unambiguous local reference frame. The Total Least Squares (TLS) estimation of the normal direction is given by EigenValue Decomposition (EVD) of the covariance matrix. The eigenvector corresponding to the smallest eigenvalue of M is defined as the normal direction. To increase repeatability in the presence of clutter, distant points are assigned smaller weights. To improve robustness to noise, all points lie within the sphericity with support of radius R. For the sake of efficiency, the centroid computation is replaced with feature point p. Therefore, covariance matrix M is calculated as a weighted linear combination:

$$\mathrm{M} = \frac{1}{\sum\limits_{i:d_i \leq R} (R - d_i)} \sum_{i:d_i \leq R} (R - d_i)\,(p_i - p)\,(p_i - p)^T,$$

$$d_{i=} \left\| p_i - p \right\|_2. \tag{9}$$

In the following, three unit eigenvectors in decreasing eigenvalue order are denoted as $x+$, $y+$ and $z+$, respectively. The opposite unit vectors are denoted as $x-$, $y-$ and $z-$, respectively. Let $\mathrm{M}(k)$ be the subset of points within the support (of radius R) whose distances from the feature point p are among the k closest to the median distance d_m, i.e.,

$$\mathrm{M}(k) = \left\{ i : |m - i| \leq k, m = \arg\,\mathrm{median} d_j \right\}. \tag{10}$$

Then, the final disambiguated x-axis is defined as:

$$S_x^+ \doteq \left\{ i : d_i \leq R \wedge \left(\mathrm{p_i} - \mathrm{p}\right) \cdot \mathrm{X}^+ \geq 0 \right\}, \tag{11}$$

$$S_x^- \doteq \left\{ i : d_i \leq R \wedge \left(\mathrm{p_i} - \mathrm{p}\right) \cdot \mathrm{X}^- > 0 \right\}, \tag{12}$$

$$S_x^+ \doteq \left\{ i : i \in M(k) \wedge \left(\mathrm{p_i} - \mathrm{p}\right) \cdot \mathrm{X}^+ \geq 0 \right\}, \tag{13}$$

$$S_x^- \doteq \left\{ i : i \in M(k) \wedge \left(\mathrm{p_i} - \mathrm{p}\right) \cdot \mathrm{X}^- > 0 \right\}, \tag{14}$$

$$\mathrm{X} = \left\{ \begin{array}{l} \mathrm{X}^+, \left|S_x^+\right| > \left|S_x^-\right| \\ \mathrm{X}^-, \left|S_x^+\right| < \left|S_x^-\right| \\ \mathrm{X}^+, \left|S_x^+\right| = \left|S_x^-\right| \wedge \left|\tilde{S}_x^+\right| > \left|\tilde{S}_x^-\right| \\ \mathrm{X}^-, \left|S_x^+\right| = \left|S_x^-\right| \wedge \left|\tilde{S}_x^+\right| < \left|\tilde{S}_x^-\right| \end{array} \right\}. \tag{15}$$

To disambiguate EVD at those points where $\left|S_x^+\right| = \left|S_x^-\right|$, it is specified that only an odd number k of vertices in $M(k)$ yield $\tilde{S}_x^+$ and $\tilde{S}_x^-$. The eigenvector is reoriented to ensure its sign is coherent with the majority of such vectors. The z axis is disambiguated by the same procedure. Finally, the y axis is given as $z \times x$.

2) SHOT DESCRIPTOR

Encoding histograms of geometric information (normal) within the support is performed by computing a set of local histograms over the 3D volumes defined by a 3D grid superimposed on the support. The grid is aligned with the axes defined by the local reference frame introduced in the previous section.

The spherical region is constructed by using query point P as the origin, using the local reference system as coordinate axes, and support R as the radius. The size of the subspace region is 32, which results from two radial divisions, two elevation divisions, and eight azimuth divisions. For each subspace, the local histograms are constructed by the following method: for each of the local histograms, the points falling into bins are accumulated according to the formula $\cos\theta_q = z_k \cdot n_q$, where n_q is the normal at the point on the local surface, z_k is the local z axis at the feature point, and θ_q is the angle between n_q and z_k.

The interval $[-1, 1]$ is selected on the horizontal axis of the histogram and is divided into bins. According to the cosine value, the corresponding histogram interval is summed. To avoid boundary effects in the process of histogram construction, when each point is accumulated into a specific local histogram bin, quadrilinear interpolation with its neighbors is performed, as shown in Fig. 10. The dimension of the histogram feature depends on the number of histogram bins. The optimal number of bins is 11, and the total descriptor length of 352 is obtained.

D. CONSTRUCTING MATCHING RELATIONS BETWEEN FRAGMENTS AND TEMPLATES

SHOT descriptors (a 352-dimensional vector) are used to describe and compare each key point on M and on F, which

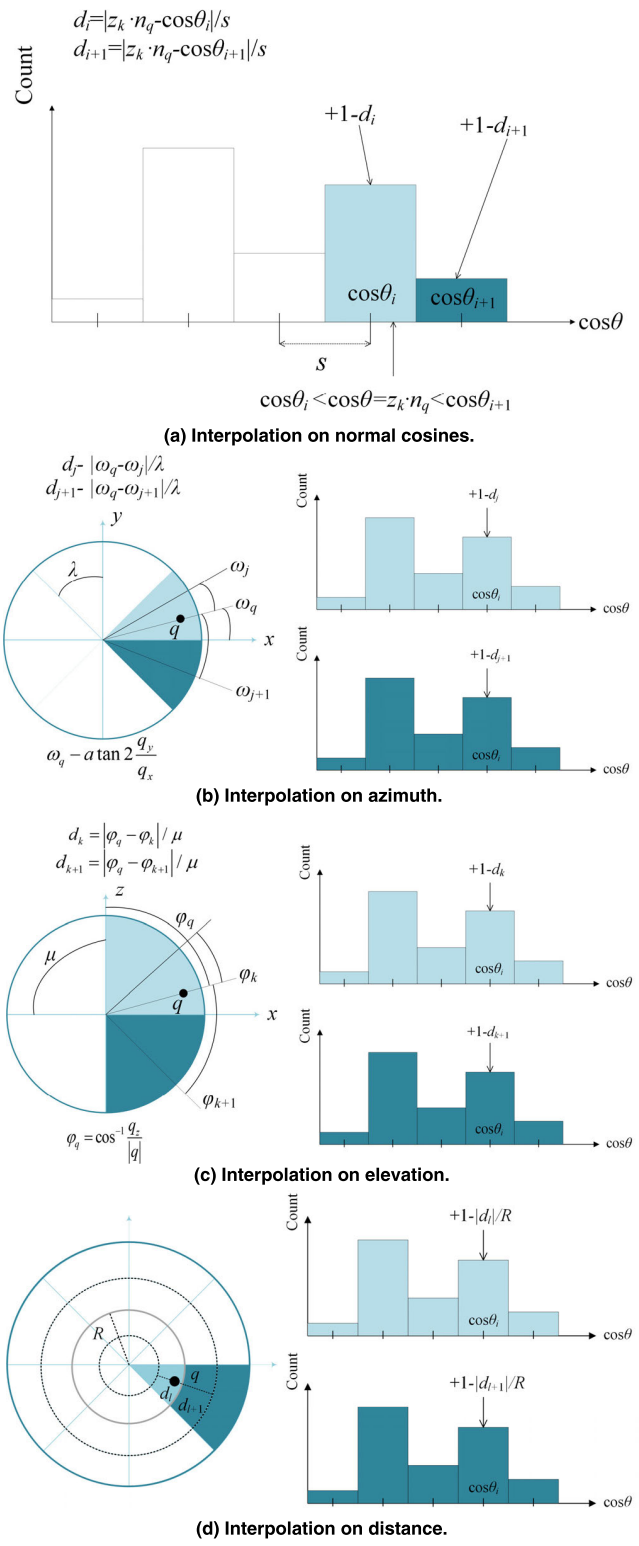


FIGURE 10. Quadrilinear interpolation to accumulate weights into histograms.

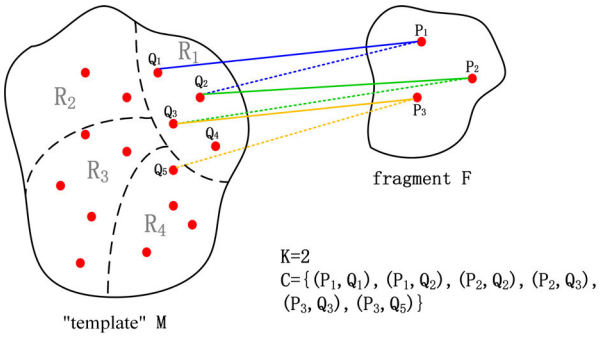


FIGURE 11. Matching relations between fragment F and template M.

are denoted as S(Q_j) and S (P_i), respectively. As shown in Fig. 11, set A = $\{Q_j | j = 1, 2, \ldots, m\}$ and B = $\{P_i | i = 1, 2, \ldots, n\}$ are the key points on template M and fragment F, respectively. If fragment F corresponds to region R_1 on template M, then for point P_i, point Q_j can be found in region R_1, making S (P_i) close to S(Q_j). A function D (S(P_i), S(Q_j) = |S(P_i) − S(Q_j)| was defined to calculate the difference between S(P_i) and S(Q_j). Here, K points on M are selected for each key point P_i on F. D(S(P_i), S(Q_{j1})) < . . . < D(S(P_i), S(Q_{jk})) ($j_1, j_2, \ldots, j_k \in$ j) is used as a constraint condition to find k smallest values. If D (S(P_i), S(Q_j)) is less than a certain threshold δs, and D (S(P_i), S(Q_j)) is one of k smallest values, (P_i, Q_j) is considered as a potential corresponding pair and added to the initial correspondence set C. When the value of K is 2, the initial correspondence set is C = {(P_1, Q_1), (P_1, Q_2), (P_2, Q_2), (P_2, Q_3), (P_3, Q_3), (P_3, Q_5)}.

E. ELIMINATING MISMATCHES BY RANSAC ALGORITHM

To ensure that enough features are extracted for fragment F, template M usually contains many key points. This makes finding the correct feature correspondence difficult. Since points from other irrelevant regions on M (that do not correspond to F) introduce irrelevant correspondences, the correct correspondence pairs are far fewer than the outliers, even though the size of the initial correspondence set was large.

From a local point of view, for each point P_i on F, (P_i,Q_f) is a pair of optimal correspondences with the minimum value of D(S(P_i), S(Q_j)) ($j = 1, 2, \ldots, f, \ldots, m$), but not from a global point of view. For example, as shown in Fig. 11, the SHOT descriptors closest to P_1, P_2, and P_3 are Q_2, Q_2, and Q_3, respectively; however, in fact {(P_1, Q_2), (P_2, Q_2), (P_3, Q_3)} is not the optimal correspondence set.

This requires that the initial correspondence set is effectively simplified and the incorrect correspondence is eliminated. The random sample consensus (RANSAC) algorithm [35] is efficient and suitable for correspondence refinement. To obtain the inliers (correct correspondence pair), we evaluate the geometric consistency among correspondence pairs given two correspondence pairs, C_1 = (P_1, Q_1) and C_2 = (P_2, Q_2), where P_1, $P_2 \in$ F and Q_1, $Q_2 \in$M. If the Euclidean distance between P_1 and P_2 is similar to the distance between Q_1 and Q_2, namely, abs($\|P_1 - P_2\| - \|Q_1 - Q_2\|$)< δ_t, we consider C_1 and C_2 to be geometrically consistent.

The steps to refine the initial correspondence set are as follows:

Step 1: Two correspondence pairs, (P_i, Q_j) and (P_m, Q_n), are randomly selected from the initial correspondence set C to construct the model. The inlier set, {(P_i, Q_j), (P_m, Q_n)}, is denoted as I. If $abs(\|P_i - P_m\| - \|Q_j - Q_n\|) > \delta_t$, we define the model to be unreasonable and select the correspondence pairs again.

Step 2: The error between each correspondence pair (P_x,Q_y) and the model is calculated. If the error satisfies $abs(\|P_x - P_i\| - \|Q_y - Q_j\|) < \delta_t$ and $abs(\|P_x - P_m\| - \|Q_y - Q_n\|) < \delta_t$, and ($P_i$, Q_j) and (P_m, Q_n) are geometrically consistent, then (P_x, Q_y) is added to inliers set I. The numbers of correspondence pairs from C and I are denoted as N and Num, respectively. If Num/N is greater than threshold δ_r, inlier set I is considered the current optimal inlier set. Otherwise, Step 1 is repeated.

Step 3: The sum (ε_I) of the errors between (P_a, Q_b) and the model is calculated according to Equation (16), where (P_a, Q_b) $\in$ I. If $\varepsilon_I < \varepsilon, \varepsilon = \varepsilon_I$. Otherwise, Step 1 is repeated.

$$\varepsilon_I = \sum_{0}^{Num} abs(\|P_a - P_i\| - \|Q_b - Q_j\|) + abs(\|P_a - P_m\| - \|Q_b - Q_n\|). \quad (16)$$

Step 4: The number of iterations *iter* is not fixed and must be updated according to Equation (17). When *iter* is greater than the greatest number of iterations, the algorithm exits, and I is the optimal correspondence set. Otherwise, the number of iterations is increased by 1, and Step 1 is repeated. If D is the confidence level, which has the value in this experiment of 0.99, then *iter* is given by:

$$iter = \frac{\log(1 - D)}{\log(1 - (\frac{Num}{N})^2)}. \quad (17)$$

Threshold δ_r is affected by two factors. One is the number of key points on template M and fragment F. The other is the value of K. We define threshold δ_r as

$$\delta_r = \lambda \frac{NUM\ (F)}{K \times NUM\ (M)}, \quad (18)$$

where λ is the weight factor and $NUM()$ represents the number of key points on the original surface of the object. To reduce mismatches, for each key point on F, we select K optimal correspondences to join the initial matching group, but only about 1/k correspondences are correct.

If the correct correspondence pairs of I reach a certain number, F and M are considered to correspond successfully. That is, the corresponding region of the fragments on the template is determined.

VII. EXPERIMENTAL RESULTS AND ANALYSIS

A. DATABASES OF TERRACOTTA WARRIOR FRAGMENTS

We obtained point cloud models of Terracotta Warrior fragments by a 3D scanner and established a fragment database that contained 878 fragments from 50 Terracotta Warriors. We labeled the archeological fragments into particular categories according to the parts of the Terracotta Warriors, some of which are shown in Fig. 12. The thickness, size, and texture

FIGURE 12. Select fragments displayed by category.

of the fragments in different parts of the Terracotta Warriors differ. For example, the head, neckline, hands, arms, legs, and feet are solid; the upper body and skirt pieces are thin and textured; and the shoulder and side sections of the body are curvy.

The total sample size was 4390 fragments after data amplification using Monte Carlo sampling. There were ten candidate classes according to the parts of the Terracotta Warriors: head, arm, hand, leg, foot, collar, the upper body and skirt, shoulder and side, and others.

B. IMPLEMENTATION DETAILS AND EXPERIMENTAL RESULTS

1) FIRST CLASSIFICATION WITH POINTNET

The fragments of the Terracotta Warriors were transformed into data in the form of a point cloud. This format is simple, and it is thus easier for deep learning methods to learn from it.

We produced point clouds with 2048 particles, where each point was a vector of its (x, y, z) coordinate plus extra normal feature channels. Each fragment data was normalized in the preprocessing step to have zero mean and unit variance.

There were 4,390 point cloud models from ten manually created categories, split into 3510 for training and 880 for testing. We train on PointNet takes 4-5 hours to converge with TensorFlow and 2 GeForce RTX 2070 GPU. The dropout with keep ratio of 0.6 on the last fully connected layer, which has an output dimension of 256, is used before the class score prediction. The decay rate for batch normalization starts with a ratio of 0.5 and is gradually raised to 0.98. PointNet uses an Adam optimizer with an initial learning rate of 0.001, momentum of 0.9, and batch size of 32.

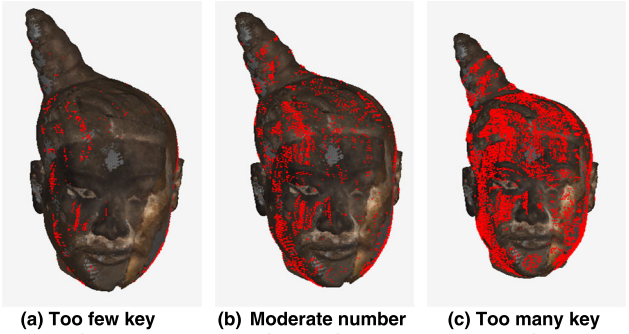
TABLE 1. Classification accuracies (%) for PointNet, PointNet++, and PointCNN.

Method	Core Operator	Accuracy (%)
PointNet	Pointwise MLP	81.79
PointNet++	Multiscale Pointwise MLP	82.01
PointCNN	χ-conv	82.15

We firstly tested our database with three models, which were directly fed to the point cloud: PointNet [23], PointNet++ [24], and PointCNN [25]. The classification accuracies of all three methods are shown in Table 1. For our dataset, accuracy varied minimally among the three deep learning networks. Therefore, we chose PointNet, which has the simplest network architecture, to classify the 3D fragments of the Terracotta Warriors.

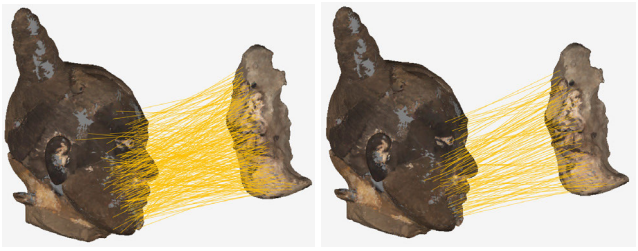
2) SECOND CLASSIFICATION GUIDED BY TEMPLATES

Our program was run on a Desktop computer with 3.6GHz Core i9 GPU and 64GB RAM. Fig. 13 shows the extraction of key points on the surface of the head template of a Terracotta Warrior when different parameter values are used. When extracting surface key points, the selection of parameters was mainly based on empirical values. As shown in Fig. 13(a), when the values of the parameter group are R $=$ 7, $\gamma_{21} = 0.75$, and $\gamma_{32} = 0.005$, too few key points are extracted from the surface of the template, resulting in too few effective matches or mismatches between key points. As shown in Fig. 13(c), when the values of the parameter group are R $= 8$, $\gamma_{21} = 0.85$, and $\gamma_{32} = 0.01$, there are too many key points to extract. This increases the complexity of generating the optimal matching group, which increases the computation time for feature descriptors. As shown in Fig. 13(b), when the values of the parameter group are R $= 8$, $\gamma_{21} = 0.9$, and $\gamma_{32} = 0.01$, the number of key points extracted is moderate.



(a) Too few key points. (b) Moderate number of key points (c) Too many key points.

FIGURE 13. Extraction of key points from the head template of the Terracotta Warriors.



(a) Initial correspondence set. (b) After deleting mismatches.

FIGURE 14. Matching of key points on fragments and templates.

Fig. 14 shows an example of the matching between the head template and the face fragments of the Terracotta Warriors. When the parameters are $\delta_s = 0.88$, k $= 4$, and $\delta_t = 7$ mm, Fig. 14(a) shows the initial matching relationship of key points between the template and the face fragment. When $\lambda = 3$ and $\delta_r = 0.342$, Fig. 14(b) shows the effect of eliminating mismatches with the RANSAC algorithm.

TABLE 2. Effects of template guidance approach on our dataset.

Method	The number of fragments	Correctly classified fragments	Accuracy (%)
PointNet	4390	3597	81.79
PointNet+template	4390	4036	90.94

Table 2 outlines the positive effects of our template guidance approach. It is interesting that PointNet achieves reasonable results. Using the templates confers a 9.15% performance boost.

As can be seen from table 3, it takes about 4 hours for the first classification of 4390 fragments with PontNet. For 793 fragments that are misclassified, it takes about 8 hours to classify them by the approach of template guidance. Our proposed method takes 11.7 hours to classify 4390 fragments, that is, an average of 9.59 seconds per fragment.

3) COMPARISON WITH OTHER CLASSIFICATION METHODS OF TERRACOTTA WARRIOR FRAGMENTS

In Table 4, we compare the accuracy of our method with a representative set of previous methods. Using the same 3D data for the fragments of the Terracotta Warriors, our method is significantly stronger than the methods used by [12] and [15]. Our method also outperforms the method used in [14].

As can be seen from the data in Table 5, it is a very time-consuming process to classify the fragments of terracotta

TABLE 3. Time consuming.

Method	The number of fragments	Time overall (h)	Time avg. (s)
PointNet	4390	3.57	2.93
Template-guide	793	8.13	36.92
PointNet+Template-guide	4390	11.7	9.59

TABLE 4. Classification accuracy (%) for proposed method and others.

Reference	Input	Sample size	Method	Accuracy (%)
[12]	image	527	SIFT+SVM	71.27
[14]	image	6000	CNN-based	89.54
[15]	volume	2096	Residual Network	83.59
This work	point cloud	4390	PointNet+ Template-guide	90.94

TABLE 5. Time consuming for proposed method and manual approach.

Method	The number of fragments	Time
Manual approach[36]	3-5	About 8 hours
This work	5	About one minute

warriors in a manual way. It's a good result to find three to five pieces of fragments that match the broken terracotta warriors in 8 hours. However, the proposed method only takes about one minute to classify 5 fragments.

C. DISCUSSION

Recently, a type of neural network that directly consumes point clouds was reported [23]–[25]. In this study, PointNet was applied to automatically classify the point cloud of the Terracotta Warrior fragments. That directly feed irregular point clouds without transforming them into regular 3D voxel grids or collections of images. PointNet achieved better classification performance on the publicly available dataset ModelNet40, a shape classification benchmark. However, for our database, the accuracy of PointNet was decreased compared to its accuracy for ModelNet40. This finding implies that the effectiveness of PointNet tends to become poor when the data are unique.

Our innovative classification method, employs Terracotta Warrior templates into the fragment classification via a second round of classification for misclassified fragments. Complete templates with geometries similar to the fragments are available and can be used to guide the classification and reassembly. It is for this reason that the proposed concept can obtain better classification performance. Our experimental results demonstrate that our hierarchical classification approach achieves significantly better performance than PointNet accomplishes by itself.

To verify the effectiveness and generality of the proposed method, we performed an analysis comparing our method to a representative set of previous methods. In terms of classification accuracy, the proposed method is higher than that of [12], [14], and [15]. Note that [12] and [14] in Table 2 used images as input by mapping a 3D model to an image. We believe that converting the 3D data into 2D images incurred unnecessary work, whereas direct use of the 3D data was the most direct approach. Liu [15] used 3D data (voxel grids) as input. However, this data representation transformation renders the resulting data unnecessarily voluminous, while also introducing quantization artifacts that can obscure natural invariances of the data. In terms of time cost, the advantages of the proposed method are obvious compared with that of manual method.

VIII. CONCLUSION

The purpose of classifying fragments of the Terracotta Warriors is to provide convenience in organizing the fragments and assisting in the subsequent reassembly tasks. Thus far, artificial and semi-automatic classification approaches are the most widely employed methods in the restoration of archaeological fragments. To improve the efficiency of the restoration of cultural relics, an automatic classification method was here is proposed. In this method, a deep learning network automatically learns the shape features of 3D fragments of the Terracotta Warriors and generates a classification of them. For those misclassified fragments, the rate of misclassification is further reduced by constructing many potential matching relationships among the fragments and their corresponding templates. The experimental results demonstrate that the accuracy of the proposed method is higher than that of any other classification method, which makes it the most suitable method for classifying archaeological fragments of the Terracotta Warriors to date. As a result, this method will significantly increase the accuracy and efficiency of future fragment reassembly of the Terracotta Warriors.

Despite the above advancements, the proposed method has limitations. For some cultural artifacts that do not have a complete template, fragment classification using only a deep learning model may not be effective. Therefore, for such cultural artifact fragments, we must identify and employ the unique and distinctive information from them and effectively combine them with deep learning methods to improve their classification accuracy.

ACKNOWLEDGMENT

The authors would like to thank every member from their group who helped collect and process the data. In addition, they would like to thank the anonymous reviewers and editor for their helpful comments.

REFERENCES

[1] M. Kampel and R. Sablatnig, "Color classification of archaeological fragments," in *Proc. IEEE 15th Int. Conf. Pattern Recognit.*, Nov. 2000, pp. 4771–4774.

[2] M. Kampel, R. Sablatnig, and E. Costa, "Classification of archeological fragments using profile primitives," in *Proc. 25th Workshop Austrian Assoc. Pattern Recognit.*, 2001, pp. 151–158.
[3] P. Smith, D. Bespalov, A. Shokoufandeh, and P. Jeppson, "Classification of archaeological ceramic fragments using texture and color descriptors," in *Proc. IEEE Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2010, pp. 49–54.
[4] L. Qi and K. Wang, "Kernel fuzzy clustering based classification of ancient-ceramic fragments," *Proc. IEEE 2nd Int. Conf. Inf. Manage. Eng.*, Apr. 2010, pp. 348–350.
[5] A. Karasik and U. Smilansky, "Computerized morphological classification of ceramics," *J. Archaeol. Sci.*, vol. 38, no. 10, pp. 2644–2657, Oct. 2011.
[6] P. Zhou, K. Wang, and W. Shui, "Ancient porcelain shards classifications based on color features," in *Proc. IEEE 6th Int. Conf. Image Graph. (ICIG)*, Aug. 2011, pp. 566–569.
[7] M. Makridis and P. Daras, "Automatic classification of archaeological pottery sherds," *J. Comput. Cultural Heritage*, vol. 5, no. 4, pp. 1–21, Jan. 2013.
[8] G. Oxholm and K. Nishino, "A flexible approach to reassembling thin artifacts of unknown geometry," *J. Cultural Heritage*, vol. 14, no. 1, pp. 51–61, Jan. 2013.
[9] N. A. Rasheed and M. J. Nordin, "Archeological fragments classification based on RGB color and texture features," *J. Theor. Appl. Inf. Technol.*, vol. 76, no. 3, pp. 358–365, 2015.
[10] R. Sablatnig, C. Menard, and W. Kropatsch, "Classification of archaeological fragments using a description language," in *Proc. IEEE 9th Eur. Signal Process. Conf.*, Sep. 1998, pp. 1–4.
[11] X. Kang, M. Zhou, and G. Geng, "Classification of cultural relic fragments based on salient geometric features," *J. Graph.*, vol. 36, no. 4, pp. 551–556, Aug. 2015.
[12] W. Yang, M. Zhou, and G. Geng, "Classification of Terra cotta warriors fragments based on multifeature and SVM," *J. Northwest Univ.*, pp. 497–504, Apr. 2017.
[13] W. Na, "Research on multi-kernel semi-supervised classification of manifold regularization based on sparse graphs," M.S. thesis, School Inf. Sci. Technol., Northwest Univ., Xi'an, China, 2018.
[14] Y. Wang, "Research on the classification algorithm of terracotta warrior fragments based on the optimization model of convolutional neural network," M.S. thesis, School Inf. Sci. Technol., Northwest Univ., Xi'an, China, 2019.
[15] X. Liu, "Research on residual network identification and multi-feature mosaic technology for cultural relics restoration," M.S. thesis, School Inf. Sci. Technol., Northwest Univ., Xi'an, China, 2019.
[16] Z. Wu, S. Song, and A. Khosla, "3D shapenets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1912–1920.
[17] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Sep. 2015, pp. 922–928.
[18] C. R. Qi, H. Su, M. Niebner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5648–5665.
[19] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953.
[20] J. Bruna, W. Zaremba, Y. LeCun, and A. Szlam, "Spectral networks and locally connected networks on graphs," Feb. 2013, *arXiv:1312.6203*. [Online]. Available: https://arxiv.org/abs/1312.6203
[21] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on riemannian manifolds," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2015, pp. 37–45.
[22] Y. Fang, J. Xie, G. Dai, M. Wang, F. Zhu, T. Xu, and E. Wong, "3D deep shape descriptor," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2319–2328.
[23] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2016, pp. 77–85.
[24] C. R. Qi, L. Yi, and H. Su, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," Jul. 2017, *arXiv:1706.02413*. [Online]. Available: https://arxiv.org/abs/1706.02413
[25] Y. Li, R. Bu, and M. Sun, "PointCNN: Convolution on $\mathcal{X}$-transformed points," Jan. 2018, *arXiv:1801.07791*. [Online]. Available: https://arxiv.org/abs/1801.07791
[26] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. Lecun, "What is the best multi-stage architecture for object recognition?" in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 2146–2153.
[27] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, no. 6789, pp. 947–951, Jun. 2000.
[28] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.
[29] K. Zhang, W. Yu, M. Manhein, W. Waggenspack, and X. Li, "3D fragment reassembly using integrated template guidance and fracture–region matching," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2138–2146.
[30] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.
[31] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3D object recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Sep. 2009, pp. 689–696.
[32] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.
[33] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.
[34] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proc. Eur. Conf. Comput. Vis. Conf. Comput. Vis.* Berlin, Germany: Springer-Verlag, 2010, pp. 356–369.
[35] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," in *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*. San Mateo, CA, USA: Morgan Kaufmann, 1987, pp. 726–740.
[36] Tencent, Shenzhen, China. (Jun. 6, 2017). *It Will Take Five Months to Restore a Terracotta Warriors*. Accessed: Nov. 7, 2019. [Online]. Available: https://news.qq.com/original/oneday/2711.html

HONGJUAN GAO received the B.S. and M.S. degrees in software and theory from Ningxia University, Yinchuan, China, in 2004 and 2007, respectively. She is currently pursuing the Ph.D. degree in computer application and technology with Northwest University, Xi'an, China. Her research interests include graph and image processing, machine learning, and computer vision. She is a member of CCF.

GUOHUA GENG received the B.S., M.S., and Ph.D. degrees in computer software and theory from Northwest University, Xi'an, China. She is currently a Professor and a Doctoral Supervisor with the School of Information Technology, Northwest University. She has presided over several National Natural Science Foundation of China projects. Her research interests include intelligent information processing and graph and image processing. She is a Senior Member of CCF.