

Received December 2, 2019, accepted December 15, 2019, date of publication December 23, 2019, date of current version January 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2961391

# Pairwise Matching for 3D Fragment Reassembly Based on Boundary Curves and Concave-Convex Patches

QUNHUI LI<sup>1</sup>, GUOHUA GENG<sup>2</sup>, AND MINGQUAN ZHOU<sup>3</sup>

<sup>1</sup>School of Sciences, Chang'an University, Xi'an 710064, China

<sup>2</sup>School of Information Science and Technology, Northwest University, Xi'an 710069, China

<sup>3</sup>College of Information Science and Technology, Beijing Normal University, Beijing 100875, China

Corresponding author: Qunhui Li (qhli@chd.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61731015.

**ABSTRACT** We present a simple pairwise matching method for 3D fragment reassembly that uses boundary curves and concave-convex patches to accelerate and optimize the matching. Given the boundary curves of two fracture surfaces that completely or partially coincide, we can quickly exclude the fracture surface pairs with low boundary curve similarities, which can substantially reduce the computational cost of the subsequent patch matching, where we extract and delineate concave-convex patches of the selected fracture surfaces. A modified iterative closest point algorithm is applied on these concave-convex patches to refine the alignment. Finally, we determine the matched 3D fragments according to the overlap ratio of their fracture surfaces. The results of experiments on real-world examples demonstrate that our proposed algorithm is both accurate and efficient.

**INDEX TERMS** Boundary curve, concave-convex patch, fracture surface, fragment matching.

## I. INTRODUCTION

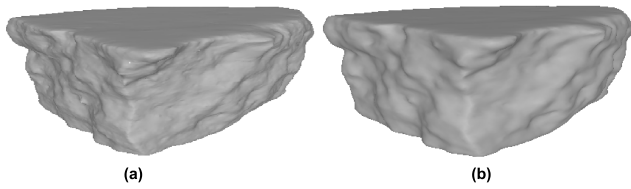
Computer-aided 3D fragment reassembly is highly important in many fields, such as archaeology, paleontology and medicine. In many cases, 3D reconstruction focuses mainly on the geometry of the fragments, since information such as color and texture is not reliable for assembly [1]. Hence, shape matching and 3D scanned digital fragment alignment are challenging problems in computer graphics and computer vision [1], [2].

Geometry-matching-based reconstruction consists of two main successive steps: pairwise matching and multi-fragment matching. The former focuses on pairwise matching of the fracture surfaces among all fragments, while the latter aims at reconstructing the entire original object according to a set of candidate matching pairs of fragments. Pairwise matching is crucial to multi-fragment matching and has been a very important technique for 3D fragment alignment [3], [4].

Recently, studies [3]–[13] matched fragments based on features of the fracture surface, which used feature patches [4], [5], feature curves [3], [6], [8], and feature points [9]. For example, Huang et al. [4] clustered points to generate

feature patches of a fracture surface and used a forward search algorithm to identify similar feature patches for pairwise matching. Wu and Wang [6] used the Hausdorff distance and the modified 4-point congruent set algorithms to identify potential simple chordless cycles for matching fragments for the reassembly of fractured sand particles. In their follow-up work [7], for matching individual fragments to their mother particle, the iterative closest point (ICP) algorithm was used to attempt to match each face of each child particle to the mother particle and to evaluate the matching results. Altantsetseg et al. [8] guided the pairwise matching of fragments by applying a descriptor that contained a cluster of feature points to the fracture surfaces and curves along the principal directions of the cluster. Others proposed non-feature-based methods for the pairwise matching of fragments. Winkelbach and Wahl [10] used a binary tree structure in a depth-first approach. Mavridis et al. [11] found an optimal match according to all vertices on the fracture surfaces and used the centroid and the normal distribution of the fracture surfaces or a standard RANSAC (Random sample consensus) for coarse alignment. Stanford's Digital Forma Urbis Romae project [12] dealt with heavily eroded fragments. Instead of using the geometry of the fracture surface, reconstruction was conducted by matching annotated

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



**FIGURE 1. Original fragment and smoothed fragment. (a) An original fragment and (b) the corresponding smoothed fragment.**

incisions on the fragments' top surfaces. In medicine, Paulano et al. [13] presented an automatic method for calculating the contact zone between two bone fragments from computed tomography (CT) images and applied the ICP algorithm to the fragment to reduce the fracture.

In this work, we combine curve matching and surface matching to improve the accuracy and efficiency of fragment matching. We directly extract the 3D boundary curve and surface features from the fracture surfaces, which are used for coarse matching and fine alignment, respectively. Our method can handle arbitrarily shaped fracture surfaces and partial matches. Our work is inspired by the work of Huang et al. [4]; however, there are several differences between our method and theirs in the aspects of fragment preprocessing, surface segmentation, feature extraction and the process of pairwise matching.

Our algorithm makes two main contributions: First, fast and simple curve matching can exclude most pairs with low surface similarities while keeping only a few candidates with high similarities. This can massively reduce the computational cost of the subsequent patch matching. Second, after extracting concave-convex patches of fracture surface pairs, a modified iterative closest point algorithm is proposed for application on the patches to optimize the assembly results. In the experiments, we evaluate the performance of the proposed algorithm on real-world examples.

In the following, we introduce two geometric features, namely, boundary curves and concave-convex patches, in Section II and Section III, respectively. In Section IV, we introduce the proposed pairwise matching approach by using the extracted features. The experimental results are presented in Section V. Finally, we present the conclusions of the paper in Section VI.

## II. BOUNDARY CURVE EXTRACTION AND COMPARISON

### A. IDENTIFYING THE FRACTURE SURFACE AND EXTRACTING THE BOUNDARY CURVE

Our target object is typically broken into several fragments, whose fracture surfaces are often bumpy; these bumps are defined as noise. Therefore, we first smooth the fragments by using adaptive smoothing [14] without shrinking, which has the advantage of controlling the degree of noise by adjusting a smoothing radius. Fig. 1 shows an original brick fragment and its smoothed version. The degree of noise of the smoothed fragment is obviously reduced, which facilitates the extraction of more accurate concave and convex patches. After smoothing, to segment the fragments into a set of surfaces

that are bounded by sharp edges, we distinguish edge points and non-edge points using multi-scale curvedness [15]. The curvedness at a point  $p$  on a surface can be estimated as:

$$c_p = \sqrt{(k_1^2 + k_2^2)/2}. \quad (1)$$

where  $k_1$  and  $k_2$  are the maximum and minimum principal curvatures, respectively. The curvedness can be used to indicate how sharp or gentle a surface is. We use the multi-scale method that is proposed in [15] to robustly estimate the curvedness by fitting a quadratic surface to neighborhood points at various scales [16]. Then, surface segmentation is realized by using a simple patch-growing algorithm [3]. Afterwards, we use the surface roughness [4] to recognize coarse fracture surfaces created by the broken object, which typically differ from the smooth and fine surfaces of the unbroken object. If the original and fracture surfaces do not differ substantially in terms of roughness, our algorithm cannot distinguish them. In this case, we must match all surfaces, including all original and fracture surfaces. However, the original surfaces can be easily excluded in the stage of pairwise matching because it is difficult to find proper surfaces that match with them.

We now have identified all fracture surfaces of the fragment; however, their boundary points are unordered and inaccurate. Thus, we use the boundary tracking algorithm [17] to obtain ordered and more accurate boundary points, which constitute the final boundary curves. Fig. 2 illustrates the extraction procedure of the fracture surface and the boundary curve of a fragment of a stone brick. Fig. 2(a) presents the multi-scale curvedness of the key points and Fig. 2(b) presents the segmentation of the fracture surface. After boundary tracking, we obtain the boundary curves, as shown in Fig. 2(c).

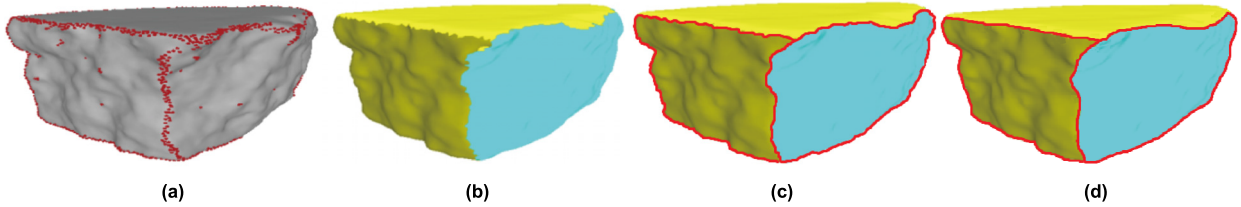
### B. BOUNDARY CURVE COMPARISON

When two fracture surfaces can be matched, their boundary curves should completely or partially coincide. Therefore, we can compare two boundary curves to judge their similarity. As illustrated in Fig. 2(c), because the extracted boundary curve includes a few noise points, the boundary curve is smoothed using a Gaussian filter to remove the noise prior to the similarity comparison. The points  $p(i)$  on the extracted boundary curve and the Gaussian filter  $f_G$  for  $p(i)$  are defined as follows:

$$p^{filt}(i) = \sum_{j=-w}^w p(i-j) \cdot f_G(j)$$

$$f_G(i) = \frac{e^{-i^2/2\sigma^2}}{\sum_{j=-w}^w e^{-j^2/2\sigma^2}}. \quad (2)$$

A filter of width  $w = 6$  performs well, and the performance at a specified filter width is related to the resolution of the boundary curve. As points with a distance of greater than  $3\sigma$  from the central point make a negligible contribution when



**FIGURE 2.** Illustration of fracture surface identification and boundary curve extraction. (a) The red points are distinguished as edge points by their curvedness. (b) The fracture surface is identified via a simple patch-growing algorithm, and each color represents a segmented surface. (c) The ordered boundary curve is obtained by boundary tracking. (d) The boundary curve after Gaussian filtering.

applying a Gaussian filter with a standard deviation of  $\sigma$ , the filter is designed using  $\sigma = w/3$ . The details are available in [3]. Fig. 2(d) shows the smoothed boundary curve after the application of the Gaussian filter.

We compare the boundary curves  $C^1$  and  $C^2$ , which are described by strings  $C^1 = \{(\kappa_1^1, \tau_1^1), \dots, (\kappa_m^1, \tau_m^1)\}$  and  $C^2 = \{(\kappa_1^2, \tau_1^2), \dots, (\kappa_n^2, \tau_n^2)\}$ , based on the discrete curvature  $\kappa$  and torsion  $\tau$ . Here,  $m$  and  $n$  are the numbers of points of  $C^1$  and  $C^2$ , respectively.

This problem is addressed as a circular substrings matching problem. This substrings matching problem is based on an  $m \times n$  similarity matrix  $\Lambda$ , in which element  $\Lambda(i, j)$  is the difference between points  $p_i^1 \in C^1$  and  $p_j^2 \in C^2$ , which is expressed as the mean Euclidean distance of  $p_i^1$  and  $p_j^2$  in the vicinity of the two points [3]. It is defined as follows:

$$\begin{aligned} \Lambda(i, j) &= \frac{1}{3} \sum_{q=-1}^1 \|p_{i+q}^1 - p_{j-q}^2\| \\ &= \frac{1}{3} \sum_{q=-1}^1 \sqrt{(\kappa_{i+q}^1 - \kappa_{j-q}^2)^2 + (\tau_{i+q}^1 - \tau_{j-q}^2)^2}. \end{aligned} \quad (3)$$

We assume that the two points are similar if  $\Lambda(i, j) < \varepsilon_\Lambda$ . A similar curve segment is a sequence of consecutive diagonal elements of  $\Lambda$ :  $\{\Lambda(i, j), \Lambda(i+1, j+1), \dots, \Lambda(i+L, j+L)\}$ , where  $L$  is the length of the detected segment.

The boundary curves of fragments that are matched partially coincide only on a small segment. We require that two similar surfaces share a boundary of at least 1/8th the length of the shortest boundary curve to avoid failing to detect the correct match.

### III. CONCAVE-CONVEX PATCH EXTRACTION AND COMPARISON

#### A. CONCAVE-CONVEX PATCH EXTRACTION

Based on the mean curvature and the Gaussian curvature at each point on a surface, local surfaces that surround the point can be subdivided into eight basic types: peak, pit, ridge, saddle ridge, valley, saddle valley, flat and minimal surface [18]. After analyzing various material fragments, we find that concave patches and convex patches mainly include pit points and peak points, respectively. The main point types between concave patches and convex patches are saddle ridges and saddle valleys.

We cluster peak points to form convex patches and cluster pit points to form concave patches. There may be several non-peak points within a group of peak points or several non-pit points within a group of pit points; we merge them to large adjacent groups. Because fracture surfaces are complicated and irregular, an ideal surface type is rarely expected. We set two thresholds, namely,  $\varepsilon_K$  and  $\varepsilon_H$ : if the Gaussian curvature satisfies  $K \geq \varepsilon_K$  and the mean curvature satisfies  $H \leq \varepsilon_H$ , the point is a peak point; if  $K \geq \varepsilon_K$  and  $H > \varepsilon_H$ , the point is a pit point.

#### B. CONCAVE-CONVEX PATCH COMPARISON

We define each patch  $R$  as follows:

$$R = \{\mu(R), \sigma(R), S(R), A(R)\}. \quad (4)$$

where  $\mu(R)$  and  $\sigma(R)$  are the mean and the standard deviation, respectively, of the curvature values of all points on patch  $R$ , which are used to represent the concavity or convexity degree of the patch  $R$ . The size signature  $S(R)$  and anisotropy signature  $A(R)$  that are associated with  $R$  are computed via principal component analysis (PCA) [19] as follows.

Let  $m$  be the total number of points on the patch  $R$  and  $H_i$  be the mean curvature of a point  $p_i \in R$ . Then,  $\mu(R)$  and  $\sigma(R)$  are respectively defined as

$$\mu(R) = \frac{1}{m} \sum_{i=1}^m H_i. \quad (5)$$

$$\sigma(R) = \sqrt{\frac{1}{m} \sum_{i=1}^m (H_i - \mu(R))^2}. \quad (6)$$

Let  $\bar{p}$  be the barycenter of patch  $R = \{p_1, \dots, p_m\}$ , and let  $M$  be a  $3 \times 3$  covariance matrix:

$$M = \sum_{i=1}^m (p_i - \bar{p})(p_i - \bar{p})^T. \quad (7)$$

where  $\lambda_0^R \geq \lambda_1^R \geq \lambda_2^R$  and  $n_0^R, n_1^R, n_2^R$  are the eigenvalues and the corresponding eigenvectors, respectively, of  $M$ .  $\lambda_i^R$  and  $n_i^R$  are also known as the principal components and the principal directions of  $R$ .  $S(R) = (\lambda_0^R + \lambda_1^R + \lambda_2^R)^{1/2}$  and  $A(R) = |\lambda_1^R / \lambda_2^R|^{1/2}$  are the size signature and the anisotropy signature, respectively. The size similarity  $SS(R_1, R_2)$  and the

anisotropy similarity  $AS(R_1, R_2)$  are defined as follows:

$$SS(R_1, R_2) = \left| \frac{S(R_1) - S(R_2)}{S(R_1) + S(R_2)} \right|. \quad (8)$$

$$AS(R_1, R_2) = \left| \frac{A(R_1) - A(R_2)}{A(R_1) + A(R_2)} \right|. \quad (9)$$

If two patches  $R_1$  and  $R_2$  satisfy the following:

$$\begin{aligned} |\mu(R_1) - \mu(R_2)| &\leq \varepsilon_\mu \\ |\sigma(R_1) - \sigma(R_2)| &\leq \varepsilon_\sigma \\ AS(R_1, R_2) &\leq \varepsilon_a \\ SS(R_1, R_2) &\leq \varepsilon_s. \end{aligned} \quad (10)$$

we claim that  $(R_1, R_2)$  is a similar patch pair.

#### IV. PROPOSED PAIRWISE FRAGMENT MATCHING

If two fracture surfaces match with each other, their boundary curves will completely or partially coincide, and the convex/concave patches of one fracture surface should match with the concave/convex patches of the paired fracture surface. In addition, since curve matching is much faster than surface matching, it can directly exclude many redundant surface pairs and reduce the high computational complexity of the subsequent concave-convex patch matching.

Our pairwise matching method consists of three steps: (a) Two fragments are roughly aligned quickly via boundary curve matching. (b) A modified ICP algorithm that is based on concave-convex patches is used for fine alignment, which is more effective than the ICP algorithm that is based on all points. (c) A validity testing strategy that is based on the overlap ratio of two fracture surfaces is applied to determine whether the two corresponding fragments match well. In our algorithm, when two fragments match successfully, both their fracture surfaces and their boundary curves match. Therefore, our algorithm yields fewer mismatches than methods that only use curve matching or surface matching.

##### A. BOUNDARY-CURVE-BASED COARSE ALIGNMENT

For boundary curves of two fracture surfaces, we employ the boundary curve comparison algorithm that was proposed in Section II and obtain all similar curve segment pairs. For each pair of similar curve segments, we calculate a transformation via the dual quaternion method [20] to coarsely align the two fragments. Next, we assign scores to all coarse alignments according to the lengths of the overlapping curve segments and store the corresponding coarse alignments in a priority queue. Starting from the head of the queue, we apply a modified ICP algorithm that integrates concave-convex patches to obtain a fine alignment.

##### B. CONCAVE-CONVEX-PATCH-BASED FINE ALIGNMENT

For fracture surface matching, in most cases, part of one surface coincides with part of the other surface. We present a modified ICP algorithm that is based on the probability iterative closest point algorithm (PICP) [21] and apply it to the concave-convex patches to obtain the fine alignment.

First, we use points on concave-convex patches instead of all surface points to find the closest points. For each point  $x_i \in R_j^1 \subset S^1$ , we find its closest point  $y_i \in R_j^2 \subset S^2$ , where patch  $R_j^2$  is the similar patch of  $R_j^1$ . If  $R_j^1$  is not similar to patch  $R_j^2$  on  $S^2$ , we discard point  $x_i$ . The similarity between patch  $R_j^1$  and  $R_j^2$  is computed according to formula (10). In this case,  $S^1$  and  $S^2$  either partially coincide or do not coincide at all. Moreover, we only use some of the points on the concave-convex patches instead of all points on the fracture surfaces for the ICP algorithm. This is sufficient for the convergence of the algorithm to the correct solution. Hence, this strategy reduces the search space for reasonable closest points and accelerates the algorithm.

Second, we use three constraints to discard outliers. The closest point pair  $(x_i, y_i)$  is identified as an outlier if one of the following constraints is satisfied: (a) the distance constraint:  $\|x_i - y_i\| > \varepsilon_d$ , namely, the distance of points  $x_i$  and  $y_i$  is larger than a threshold  $\varepsilon_d$ ; (b) the normal constraint:  $\|n(x_i) - n(y_i)\| > \varepsilon_n$ , namely, the difference between the normals at points  $x_i$  and  $y_i$  exceeds a threshold  $\varepsilon_n$ ; and (c) the concavity-convexity degree constraint:  $|H(x_i) - H(y_i)| > \varepsilon_c$ , where  $H(\cdot)$  is the mean curvature, namely, the concavity-convexity degree differs substantially between points  $x_i$  and  $y_i$ .

The method of removing outliers enables us to realize part-part surface matching. Using the three constraints to remove outliers, we can obtain the closest point pairs with fewer outliers. The closest point pairs are used to compute the transformation, which yields an accurate estimate and accelerates the convergence.

##### C. VALIDITY TESTING STRATEGY

The validity testing strategy is based on the overlapping area ratio of two fracture surfaces. The area of a surface is described by the number of points on it. According to our experimental analysis, when the fracture surfaces matched correctly, the minimum overlapping percentage between the surfaces was approximately 20%. Hence, 20% is set as the threshold for successful matching.

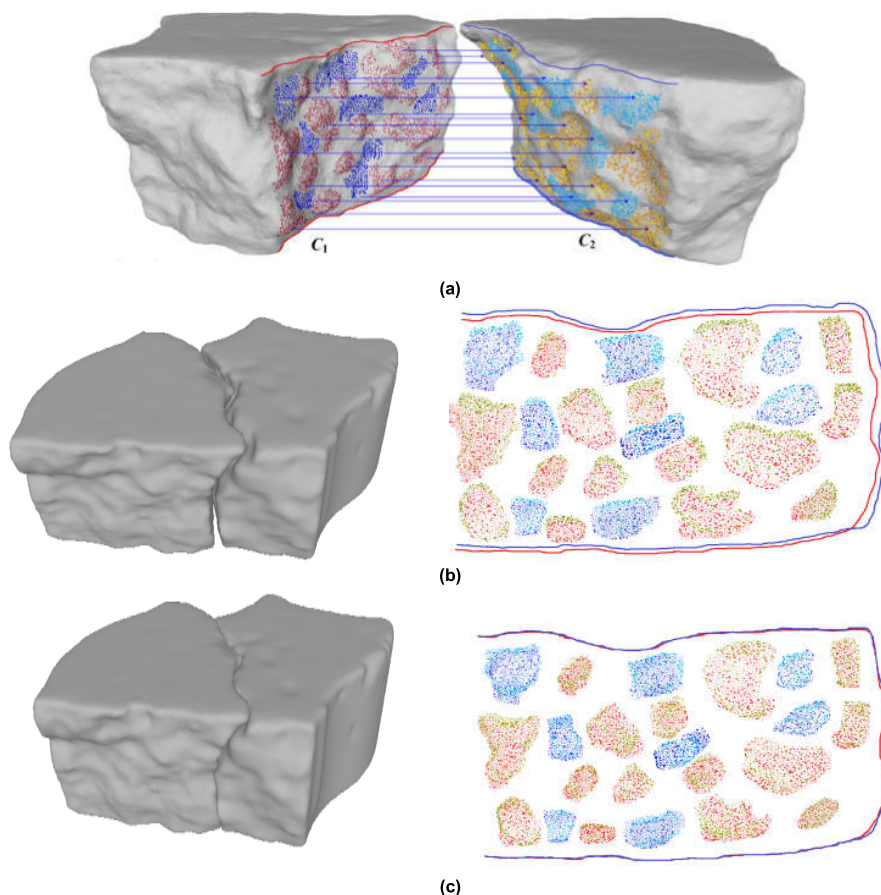
Therefore, we consider two fracture surfaces  $S^1$  and  $S^2$  to be correctly matched or to fit well if the size of their overlapping area is larger than 20% of the area of one of the fracture surfaces [19].

To define the overlapping area between surface  $S^1$  and  $S^2$ , we consider points from both  $S^1$  and  $S^2$  and their closest points on another surface. Considering a point  $x_i$  from  $S^1$  as an example, we define  $y_i$  in the overlapping area between  $S^1$  and  $S^2$  if it satisfies:

$$\begin{aligned} \|x_i - y_i\| &\leq \varepsilon_{dis} \\ \|n(x_i) - n(y_i)\| &\leq \varepsilon_\theta \\ |H(x_i) - H(y_i)| &\leq \varepsilon_{curv}. \end{aligned} \quad (11)$$

where  $y_i$  is the closest point of  $x_i$  on  $S^1$ ,  $n(\cdot)$  denotes the normal and  $H(\cdot)$  is the mean curvature.





**FIGURE 3.** Pairwise matching demonstration of two fragments. (a) Two similar fractures and their boundary curves and concave-convex patches. (b) Coarse alignment via boundary curves; (c) Fine alignment via concave-convex patches.

In the implementation, we find that even if a pair of incorrectly matched fracture surfaces have several consistent point correspondences, their overlapping fracture surface is small. Therefore, most incorrect fracture surface matches are eliminated by the validity test.

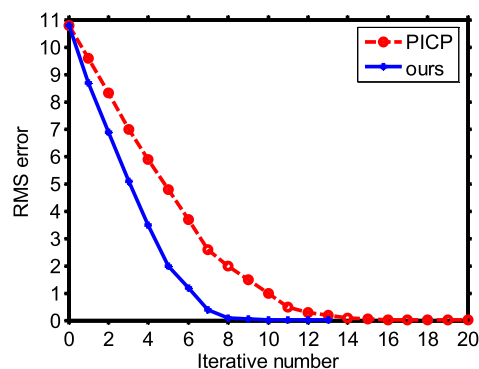
### V. EXPERIMENTAL RESULTS

We applied the proposed algorithm to several real-world broken objects of various materials, which included brick (stone) and cake (mortar) models from Vienna University of Technology [4]. The brick and cake fragments are presented in [22]. The triangular meshes of the head (terracotta) fragments are obtained using a 3D laser scanner, namely, Handy3dscan. All fragments are scanned from real fractured models. The experiment was conducted on a computer with a Pentium(R)/3.4 GHz CPU and 8.0 GB of RAM.

#### A. THRESHOLD PARAMETER SETTING

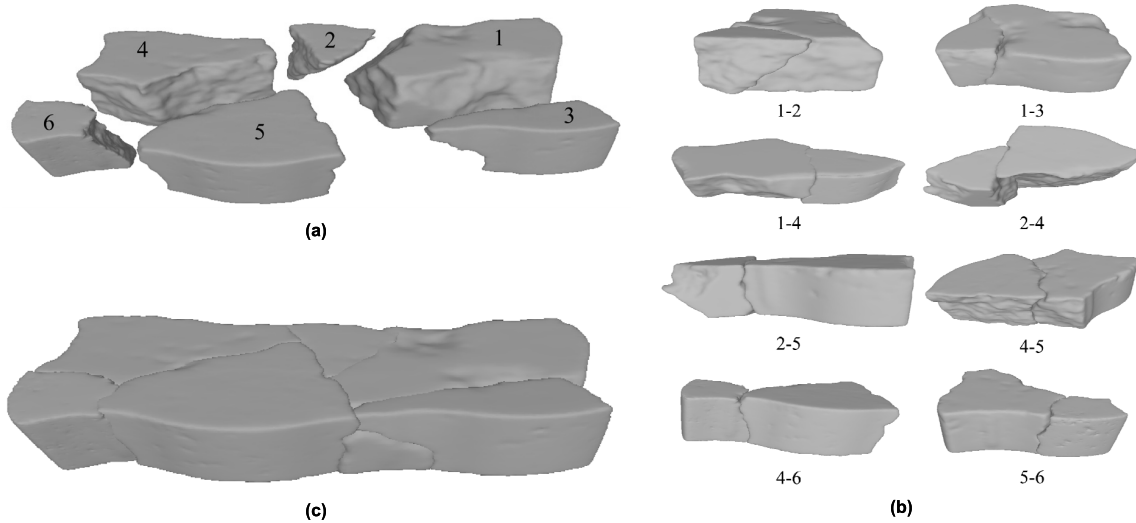
Parts of our algorithm are guided by thresholds. Following references [4] and [19], we use statistical methods to estimate the thresholds. According to the attributes of the thresholds, we classify the thresholds into two categories.

The thresholds in the first category are used in the similarity comparison of surfaces. They include mainly  $\epsilon_\Lambda$  for

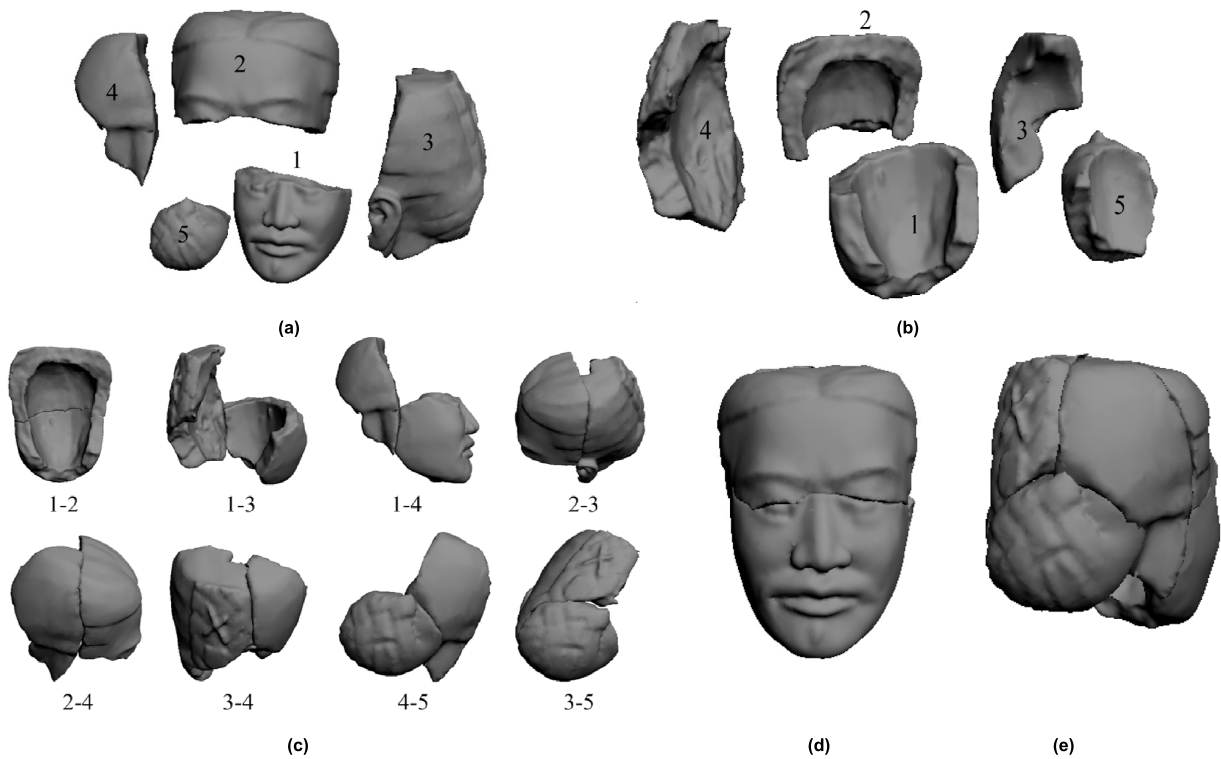


**FIGURE 4.** Convergence processes of PICP and the modified ICP on two fragments in Fig. 3.

boundary curve similarity;  $\epsilon_\mu$ ,  $\epsilon_\sigma$ ,  $\epsilon_a$ ,  $\epsilon_s$  in Eq. (10) for concave-convex patch similarity; and  $\epsilon_{dis}$ ,  $\epsilon_\theta$ ,  $\epsilon_{curv}$  in Eq. (11) for validity testing. To prevent failure to detect the correct match, these thresholds are set to large values. For false matches, the corresponding error is much larger than these thresholds. We set  $\epsilon_\Lambda = 0.3$ ,  $\epsilon_\mu = 0.3$ ,  $\epsilon_\sigma = 0.2$ ,  $\epsilon_a = 0.1$ ,  $\epsilon_s = 0.1$ ,  $\epsilon_{dis} = 0.01$ ,  $\epsilon_\theta = 0.01$  and  $\epsilon_{curv} = 0.03$ .



**FIGURE 5.** Brick fragments and their pairwise matching results, and the global reassembly result. (a) Six brick fragments; (b) The pairwise matching results; (c) The reassembly result.



**FIGURE 6.** Broken fragments, their pairwise matching results, and the global reassembly result. (a) and (b) Two views of five head fragments; (c) The pairwise matching results. (d) and (e) Two views of the global reassembly result.

The thresholds in the second category include  $\epsilon_K$  and  $\epsilon_H$  for concave-convex patch extraction and  $\epsilon_d, \epsilon_n$  and  $\epsilon_c$  for discarding outliers when finding the closest point pairs. We use all fragments to estimate and analyze these thresholds. When  $\epsilon_K$  and  $\epsilon_H$  are set in the interval  $[0.001, 0.01]$ , the difference in the results of concave and convex region extraction is small. Thus, we set  $\epsilon_K = 0.005$  and  $\epsilon_H = 0.005$ . We found that large values of  $\epsilon_d, \epsilon_n$  and  $\epsilon_c$ , which are used to prune the false point pairs, do not greatly influence the final results but do affect the running time of the ICP algorithm. Therefore, in

our experiments,  $\epsilon_d, \epsilon_n$  and  $\epsilon_c$  are respectively set as  $\epsilon_d = \frac{5}{N} \sum_{i=1}^N \|x_i - y_i\|$ ,  $\epsilon_n = \frac{5}{N} \sum_{i=1}^N \|n(x_i) - n(y_i)\|$  and  $\epsilon_c = 0.3$ , where  $N$  is the total number of closest point pairs of the two fracture surfaces in the iteration.

**B. ILLUSTRATION AND CONVERGENCE ANALYSIS OF THE PROPOSED PAIRWISE MATCHING APPROACH**

Fig. 3 presents the matching results of boundary curves and fracture surfaces before and after the application of

**TABLE 1. Comparison between PICP and our proposed method on the brick fragments.**

Brick fragments	Triangular faces		RMS error (mm)		Number of iterations		Time (s)	
			PICP	Ours	PICP	Ours	PICP	Ours
1-2	418,934	278,234	0.0156	0.0143	21	13	4.3	2.5
1-3	418,934	279,204	0.0148	0.0136	16	10	3.9	1.9
1-4	418,934	415,206	0.0139	0.0130	20	12	4.1	2.3
2-4	278,234	415,206	0.0102	0.0081	16	9	3.7	1.9
2-5	278,234	416,866	0.0112	0.0094	15	8	3.5	1.8
4-5	415,206	416,866	0.0153	0.0141	20	13	4.2	2.2
4-6	415,206	277,290	0.0131	0.0122	16	10	4.0	2.0
5-6	416,866	277,290	0.0114	0.0103	20	13	4.3	2.4

**TABLE 2. Comparison between PICP and our proposed method on the head fragments.**

Head fragments	Triangular faces		RMS error (mm)		Number of iterations		Time (s)	
			PICP	ours	PICP	Ours	PICP	Ours
1-2	50,204	35,368	0.0189	0.0181	29	17	3.1	1.7
1-3	50,204	33,026	0.0177	0.0165	28	15	3.0	1.7
1-4	50,204	14,566	0.0157	0.0147	26	14	2.9	1.6
2-3	35,368	33,026	0.0203	0.0193	32	18	3.1	1.8
2-4	35,368	14,566	0.0245	0.0234	36	21	3.3	2.0
3-4	33,026	14,566	0.0198	0.0188	31	17	3.1	1.7
4-5	14,566	8,356	0.0186	0.0178	29	16	3.1	1.7
3-5	33,026	8,356	0.0243	0.0232	33	18	3.3	1.9

our ICP algorithm. Fig. 3(a) shows two similar fractures and their boundary curves and concave-convex patches. The coarse alignment results of two fragments, boundary curves and concave-convex patches via similar curve segment pair ( $C_1$ ,  $C_2$ ) in Fig. 3(a) are present in Fig. 3(b), and the fine alignment results that are obtained using our ICP algorithm based on similar concave-convex patches are presented in Fig. 3(c).

Fig. 4 presents the convergence processes of PICP [21] and our modified ICP on two fragments that are shown in Fig. 3. The root mean square (RMS) error [21] is used to measure the residual error in registration algorithms. Our modified ICP method converges faster than PICP under the same convergence condition.

### C. PAIRWISE MATCHING AND GLOBAL ASSEMBLY VIA THE PROPOSED METHOD FOR REAL FRAGMENTS

Fig. 5 shows six brick fragments (Fig. 5(a)), their pairwise matching results (Fig. 5(b)), and the global reassembly result (Fig. 5(c)) that is based on pairwise matching. Similarly, Fig. 6 shows a matching example of a head model with five fragments.

In this work, we focus mainly on the pairwise matching of fragments. Based on our pairwise matching, we use the simple and greedy method in [5] to globally reassemble fragments. It is used to evaluate the performance of our method. Based on the proposed pairwise matching method, we can obtain all fracture surface pairs of all fragments for the subsequent complete reconstruction. The global assembly process is as follows: First, the fracture surface pair with the maximum overlap region is selected and assembled after calculating the related transformation matrix. This process is repeated for the remaining fracture surface pairs of unassembled fragments until all fragment pairs have been assembled.

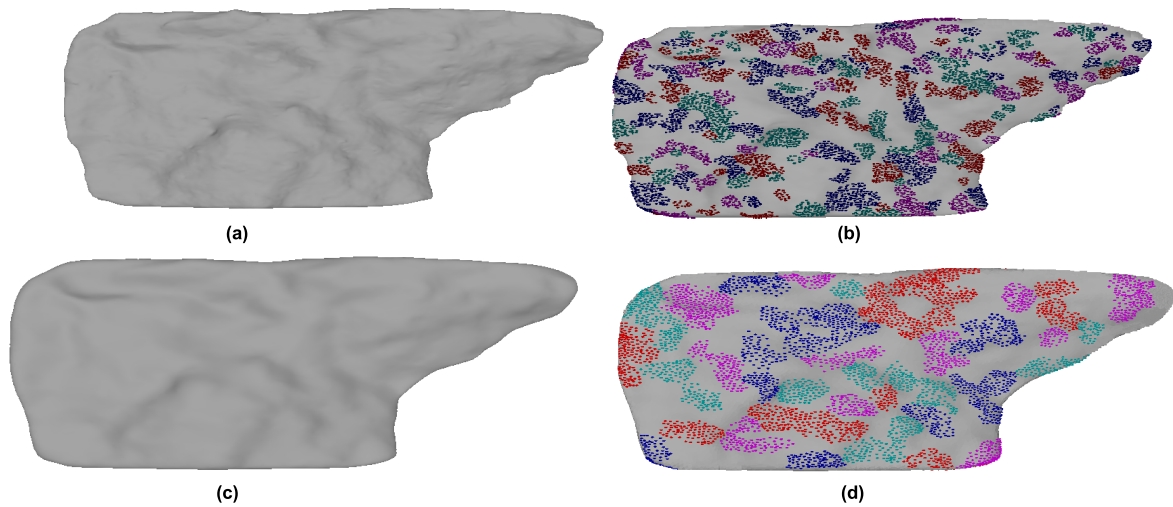
For the brick and head fragments, we can find all potential matches. A complete match is a match in which two fracture surfaces coincide entirely. The remaining matches are partial matches, in which only part of the surface coincides with all or part of the other surface. The related experimental results in Fig. 5(b) and Fig. 6(c) demonstrate that our method can realize the complete or partial matching of broken objects.

Tables 1–2 compare the performances of the PICP algorithm [21] and our algorithm on the brick and head fragments that are presented above in terms of the RMS error, number of iterations and speed. The proposed method outperforms the PICP algorithm [21] with fewer iterations and higher execution speed under the same convergence condition. The accuracy of our algorithm is also slightly higher than that of the PICP algorithm [21].

### D. ABLATION STUDY OF THE PROPOSED METHOD

Fig. 7 and Table 3 present an experimental comparison between the original surfaces and the smoothed surfaces of the brick fragments. From the original fragment in Fig. 7(a), 131 concave-convex patches (Fig. 7(b)) were extracted. Most of these patches are too small to accurately reflect the concavity and convexity characteristics of the fracture surface. Then, 34 concave-convex patches (Fig. 7(d)) were extracted from the smoothed fragment in Fig. 7(c).

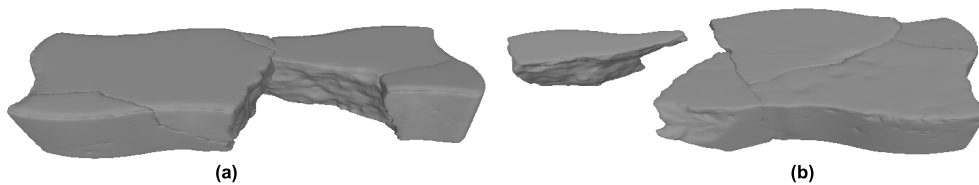
Compared with the original fragments, the speeds of the algorithms on the smoothed fragments increased by approximately 18.5%. For the smoothed fragments, we can find all potential matches, whereas original brick fragments 2-5 were not matched successfully. In Table 3, for those successfully matched fragment pairs, the average RMS errors of smooth fragments is approximately 0.078 of the error of the original fragments with the same number of iterations.



**FIGURE 7.** Comparisons of original fragment and smoothed fragment in concave-convex patches. (a) Original fragment; (b) The concave-convex patches that were extracted from the original fragment; (c) The corresponding smoothed fragment of (a); (d) The concave-convex patches that were extracted from the smoothed fragment.

**TABLE 3.** Comparison between the original surface and the smoothed surface for the matching of brick fragments.

fragment pairs	Number of concave-convex patches		Matching time (s)		RMS error (mm)		Matching result			
	original fragment	smoothed fragment	original fragment	smoothed fragment	original fragment	smoothed fragment	original fragment	smoothed fragment		
1-2	136	145	28	31	15	12	0.2011	0.0143	success	success
1-3	221	236	42	45	21	17	0.1789	0.0136	success	success
1-4	186	174	32	34	19	16	0.1267	0.0130	success	success
2-4	177	174	28	34	19	16	0.1103	0.0081	success	success
2-5	177	91	28	16	15	11	2.1105	0.0094	failure	success
4-5	190	161	36	33	17	14	0.1985	0.0141	success	success
4-6	190	56	36	11	13	11	0.1567	0.0122	success	success
5-6	95	100	18	19	11	9	0.1211	0.0103	success	success



**FIGURE 8.** Reassembly results with a piece of a fragment missing. (a) The reassembly result with brick 5 missing; (b) The reassembly result with brick 1 missing.

Although fewer concave-convex patches were extracted based on the smooth fragments, these patches can more robustly describe the concave and convex shapes and characteristics of the fracture surface. Furthermore, because the search space can be reduced and the matching speed can be increased, higher accuracy is realized.

In both archeological and medical applications, it is very common for some fragments to be unavailable. If a piece of a fragment is missing, our method can still find all potential fragment pairs in the remaining fragments. However, the global reassembly results are affected. For example, Fig. 8(a) presents the reassembly results with brick 5 missing. In this case, the absence of brick 5 will not affect the assembly

result of the remaining fragments. Fig. 8(b) presents the reassembly results with brick 1 missing, in which brick 3 is isolated because it only pairs with brick 1.

**VI. CONCLUSION**

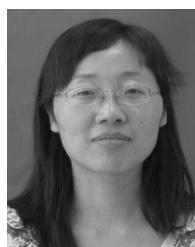
In this paper, we present a simple but effective method for the pairwise matching of 3D fragments, which consists of boundary-curve-based coarse matching and concave-convex-patch-based fine alignment. First, we compare the fracture surfaces' boundary curves of 3D fragments that have high similarities, which are regarded as matched surfaces. This simple boundary-curve-based comparison can facilitate fast coarse matching to quickly exclude low-similarity boundary



curves from surface pairing. Next, based on the results of the previous step, we conduct fine alignment according to concave-convex patches of the selected fracture surfaces. Afterwards, we determine whether two fragments fit well based on their overlap ratio. The experimental results demonstrate that our method realizes efficient and accurate matching for various types of fragments.

## REFERENCES

- [1] N. A. Rasheed and M. J. Nordin, "A survey of computer methods in reconstruction of 3D archaeological pottery objects," *J. Adv. Res.*, vol. 3, no. 3, pp. 712–724, 2015, doi: [10.1117/12.420923](https://doi.org/10.1117/12.420923).
- [2] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, "3D object recognition in cluttered scenes with local surface features: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2270–2287, Nov. 2014, doi: [10.1109/TPAMI.2014.2316828](https://doi.org/10.1109/TPAMI.2014.2316828).
- [3] G. Papaioannou and E. A. Karabassi, "On the automatic assemblage of arbitrary broken solid artefacts," *Image Vis. Comput.*, vol. 21, no. 5, pp. 401–412, 2003, doi: [10.1016/s0262-8856\(03\)00008-8](https://doi.org/10.1016/s0262-8856(03)00008-8).
- [4] Q.-X. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann, "Reassembling fractured objects by geometric matching," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 569–578, Aug. 2006, doi: [10.1145/1141911.1141925](https://doi.org/10.1145/1141911.1141925).
- [5] T. Son, J. Lee, J. Lim, and K. Lee, "Reassembly of fractured objects using surface signature," *Vis. Comput.*, vol. 34, no. 10, pp. 1371–1381, 2018, doi: [10.1007/s00371-017-1419-0](https://doi.org/10.1007/s00371-017-1419-0).
- [6] M. Wu and J. Wang, "Reassembling fractured sand particles using fracture-region matching algorithm," *Powder Technol.*, vol. 338, pp. 55–66, Oct. 2018, doi: [10.1016/j.powtec.2018.06.045](https://doi.org/10.1016/j.powtec.2018.06.045).
- [7] M. Wu and J. Wang, "Registration of point cloud data for matching crushed sand particles," *Powder Technol.*, vol. 347, pp. 227–242, Apr. 2019, doi: [10.1016/j.powtec.2019.03.001](https://doi.org/10.1016/j.powtec.2019.03.001).
- [8] E. Altantsetseg, K. Matsuyama, and K. Konno, "Pairwise matching of 3D fragments using fast Fourier transform," *Vis. Comput.*, vol. 30, nos. 6–8, pp. 929–938, 2014.
- [9] E. Vendrell-Vidal and C. Sánchez-Belenguier, "A discrete approach for pairwise matching of archaeological fragments," *ACM J. Comput. Cultural Heritage*, vol. 7, no. 3, pp. 1–19, 2014, doi: [10.1145/2597178](https://doi.org/10.1145/2597178).
- [10] S. Winkelbach and F. M. Wahl, "Pairwise matching of 3D fragments using cluster trees," *Int. J. Comput. Vis.*, vol. 78, no. 1, pp. 1–13, 2008, doi: [10.1007/s11263-007-0121-5](https://doi.org/10.1007/s11263-007-0121-5).
- [11] P. Mavridis, A. Andreadis, and G. Papaioannou, "Fractured object reassembly via robust surface registration," in *Proc. EG*, 2015, pp. 21–24.
- [12] D. Koller and M. Levoy, "Computer-aided reconstruction and new matches in the forma urbis romae," *Bullettino Della Commissione Archeol. Comunale Roma*, vol. 2, no. 15, pp. 103–125, 2006.
- [13] F. Paulano-Godino and J. J. Jiménez-Delgado, "Identification of fracture zones and its application in automatic bone fracture reduction," *Comput. Methods Programs Biomed.*, vol. 141, pp. 93–104, Apr. 2017, doi: [10.1016/j.cmpb.2016.12.014](https://doi.org/10.1016/j.cmpb.2016.12.014).
- [14] G. Lavoué, "A roughness measure for 3D mesh visual masking," in *Proc. APGV*, Tübingen, Germany, 2007, pp. 57–60.
- [15] H. T. Ho and D. Gibbins, "Curvature-based approach for multi-scale feature extraction from 3D meshes and unstructured point clouds," *IET Comput. Vis.*, vol. 3, no. 4, pp. 201–212, Dec. 2009, doi: [10.1049/iet-cvi.2009.0044](https://doi.org/10.1049/iet-cvi.2009.0044).
- [16] B. Hamann, "Curvature approximation for triangulated surfaces," *Computing*, vol. 8, no. 8, pp. 139–153, 1993, doi: [10.1007/978-3-7091-6916-2\\_10](https://doi.org/10.1007/978-3-7091-6916-2_10).
- [17] S. Liu, "Research on key technology in reconstruction of free-form & regular surfaces in reverse engineering," Ph.D. dissertation, Dept. Mech. Elect. Eng., NUAU, Nanjing, China, 2004.
- [18] P. J. Besl and R. C. Jain, "Invariant surface characteristics for 3D object recognition in range images," in *Proc. CVGIP*, 1986, pp. 33–80.
- [19] Q. Huang and H. Pottmann, "Automatic and robust multi-view registration," Geometry Preprint Series, Vienna Univ. Technol., Vienna, Austria, Tech. Rep. 152, 2005.
- [20] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternion," *Opt. Soc. Amer.*, vol. 44, no. 4, pp. 629–642, 1987, doi: [10.1364/JOSAA.4.000629](https://doi.org/10.1364/JOSAA.4.000629).
- [21] S. Du, J. Liu, C. Zhang, J. Zhu, and K. Li, "Probability iterative closest point algorithm for m-D point set registration with noise," *Neurocomputing*, vol. 157, pp. 187–198, Jun. 2015, doi: [10.1016/j.neucom.2015.01.019](https://doi.org/10.1016/j.neucom.2015.01.019).
- [22] *3D Puzzles-Reassembling Fractured Objects by Geometric Matching*. Accessed: Aug. 20, 2018. [Online]. Available: <http://www.geometric.tuwien.ac.at/ig/3dpuzzles.html>



**QUNHUI LI** received the B.S. degree in computer science and technology from Sichuan University, Chengdu, China, in 1997, and the M.S. and Ph.D. degrees in computer software and theory from Northwest University, Xi'an, China, in 2005 and 2013, respectively. Since 2009, she has been a Lecturer with the School of Sciences, Chang'an University, Xi'an. Her research interests include computer graphics, image processing, and the digital restoration of cultural heritage.



**GUOHUA GENG** received the Ph.D. degree in computer software and theory from Northwest University, Xi'an, China, in 2004. Since 2002, she has been a Professor with the School of Information Science and Technology, Northwest University. Her research interests include image processing, intelligent information processing, and the digital restoration of cultural heritage.



**MINGQUAN ZHOU** is currently a Professor with the School of Information Science and Technology, Northwest University, and the College of Artificial Intelligence, Beijing Normal University, a Ph.D. Supervisor, and an Outstanding Member of the Chinese Computer Society. His main research interests include virtual reality, image processing, visualization technology, and so on.