Automatic Reassembly Method of 3D Thin-wall Fragments Based on Derivative Dynamic Time Warping

Siyong Fu[(1)]**, Lushen Wu**[(2)]**, Qiuxiang Tao**[(1)]**, Ying Shen**[(1)]

[(1)] *ZTE School of Communication and Information Engineering, Xinyu University, Xinyu 338004, China*
    e-mail: fusiyong58@163.com

[(2)] *School of Mechanical and Electrical Engineering, Nanchang University, Nanchang 330031, China*

SUMMARY

In order to address the automatic virtual reassembling of 3D thin-wall fragments, this paper proposes a 3D fragment reassembly method based on derivative dynamic time warping. Firstly, a calculation method of discrete curvature and torsion is designed to solve the difficulty of calculating curvature and torsion of discrete data points and eliminate effectively the noise interferences in the calculation process. Then, it takes curvature and torsion as the feature descriptors of the curve, searches the candidate matching line segments by the derivative dynamic time warping (DDTW) method with the feature descriptors, and records the positions of the starting and ending points of each candidate matching segment. After that, it designs a voting mechanism with the geometric invariant as the constraint information to select further the optimal matching line segments. Finally, it adopts the least squares method to estimate the rotation and transformation matrices and uses the iterative closest point (ICP) method to complete the reassembly of fragments. The experimental results show that the reassembly error is less than 1mm and that the reassembly effect is good. The method can solve the 3D curve matching in case there are partial feature defects, and can achieve the virtual restoration of the broken thin-wall fragment model quickly and effectively.

KEY WORDS: *derivative dynamic time warping; virtual restoration; discrete curvature; discrete torsion; fragment reassembly.*

1. INTRODUCTION

In each archaeological excavation process, a large number of broken cultural relics would be unearthed. The conventional method is to reassemble the fragments by manual splicing. However, when the number of fragments is too large, this manual method will be inefficient, costly, and even lead to splicing failure due to the small size or inconspicuous structural features of some fragments. Therefore, the reassembly of 3D fragments has far-reaching, theoretical and application values [1-2].

Generally, space fragments can be divided into two categories: one is the space thin-wall fragments, with small fracture surface areas and less feature information for matching, and the assembly is mainly based on the contour curve information of the fragments; and the other is thick-wall fragments, with large fracture surface areas, and the assembly mainly takes into account the matching between surfaces. This paper focuses on the reassembly of thin-wall fragments and studies the matchability between 3D curves.

Many scholars have studied spatial curve matching. Wolfson et al. [3] re-sampled the spatial curves and calculated the curvatures of the sample points, encoded the curvature information of the sample points, by matching them with the hash algorithm. The results show that the matching accuracy is related to the sampling frequency. Leitão et al. [4] encoded the curvature information of the vertices of the curves and used the dynamic programming algorithm to compare the encoded information to further determine the matching relationship between curves. Ucoluk et al. [5] composed the feature parameter vectors based on the curvatures and torsions at the vertices of the contour curves and transformed the spatial curve matching problem into a simple feature string matching problem. The curvature and torsion information of the vertices are calculated by the differential method, but the precondition is that the points on the curves should be evenly distributed. However, in the actual situation, the points on the contour curves are discretely and randomly distributed, and the exhaustive method is adopted in the matching process, which increases the time complexity. Oxholm et al. [6] combined the vertex curvature, torsion, and colour to form feature descriptors, and used the longest common subsequence (LCS) method to find the optimum neighbour relationship between curves. This method can well address the reassembly of fragments with colour matching. However, the LCS algorithm requires that the spatial density of the two sequence data be consistent, and what is more, a large number of cultural relic fragments have lost colour characteristics due to erosion by chemicals, wind, and rain [7]. Zhang et al. [8] proposed a method for reassembling fractured fragments based on skeleton graphs matching, which transforms the problem of fragment matching into how to match the texture information of the fragment surfaces, and organically combined the texture information of the surfaces with the contour information, and searched the matching fragments using a certain searching method. This method can solve well the matching between fragments with regular textures. All of the above methods require that the features to be matched should be perfect without any missing part, but in actual practice, relic fragments often have either missing features or missing parts. Using directly any of the above methods will not help to obtain good results. For these partially defective fragments, some domestic scholars proposed a semi-automatic reassembly method. Li Shanshan and Zhou Mingquan, et al. [9-11] proposed this method. Firstly, it gives the matching feature point pairs according to the subject matter experts, then has the feature point pairs locally grown to produce new ones, and finally determines the adjacency relationship between the contour lines according to the angles and point distances between feature point pairs. However, the performance of the semi-automatic reassembly method depends heavily on the feature point pairs given by the subject matter experts.

In order to address the automatic reassembling of 3D thin-wall fragments, this paper proposes an automatic reassembly method based on DDTW. Firstly, this paper designs a calculation method for discrete curvature and torsion then takes curvature and torsion as the feature-invariant descriptors to search the candidate matching line segments by the DDTW method and records the positions of the starting and ending points of each candidate matching segment. According to the geometric invariant constraints of the points between candidate matching segments, it eliminates the mismatching and further on selects the optimal matching

line segments. Finally, it adopts the least squares method to estimate the rotation and transformation matrices and uses the ICP [12] method to complete the reassembly of fragments.

2. CURVE SHAPE DESCRIPTION BASED ON DISCRETE CURVATURE AND TORSION

The fragment reassembly is based on the valid feature shape descriptors. The feature information should have spatial rotation and translation invariance. According to the knowledge of differential geometry, the curvature and torsion of a point on the curve are geometric invariants, that is, they are invariant with the parameter variations [13]. So curvature and torsion can be used as the shape descriptors of curves. The curvature and torsion at any point on a space curve can be expressed as [14]:

$$k(t) = \frac{|r'(t) \times r''(t)|}{|r'(t)|^3} \tag{1}$$

$$\tau(t) = \frac{(r'(t) \times r''(t)) \cdot r'''(t)}{|r'(t) \times r''(t)|^2} \tag{2}$$

where, k is the curvature, and τ is the torsion, so the shape of any point P_i on the curve can be represented by the vector $s[k_i, \tau_i]$. From equations (1) and (2), it can be seen that the calculation of curvature and torsion requires a known curve table and that the curve needs to be the third derivative. However, the actually extracted contour profile is discrete, which means it cannot be calculated directly with the above formulas. Following the idea of reference [15], this paper proposes a method for estimating curvature and torsion based on discrete geometric information.
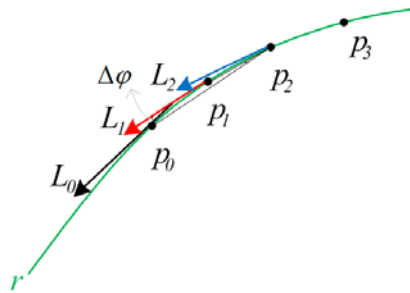

Fig. 1 *Schematic diagram of curvature estimation*

4 adjacent points P_0, P_1, P_2, P_3 are selected on the space curve, and the tangent lines P_0L_0, P_1L_1, P_2L_2 are drawn across the points P_0, P_1, P_2. Let $l(P_i, P_j)$ be the arc length from point P_i to P_j along the curve r, and suppose that $l(P_0, P_1) = l(P_1, P_2) = \Delta l$, $l(P_0, P_2) \approx 2\Delta l$ and $\Delta\varphi = \angle(P_0L_0, P_1L_1) = \pi - \angle(P_2P_0, P_1P_2)$, as shown in Figure 1. When $\Delta l \to 0$, $\Delta\varphi \to 0$, and then:

$$k_{p_1} = \lim_{\Delta l = l(P_0, P_1) \to 0} \frac{\Delta\varphi}{\Delta l} = \lim_{\Delta s \to 0} \frac{2\Delta\varphi}{|P_0P_2|} \cdot \frac{|P_0P_2|}{l(P_0, P_2)} = \lim_{\Delta\varphi \to 0} \frac{2\sin\Delta\varphi}{|P_0P_2|} =$$

$$\lim_{\Delta\varphi \to 0} \frac{2}{|P_0P_2|} \cdot \frac{|P_1P_0 \times P_1P_2|}{|P_1P_0||P_1P_2|} = \lim_{\Delta\varphi \to 0} \frac{2|P_1P_0 \times P_1P_2|}{|P_0P_2||P_1P_0||P_1P_2|} \tag{3}$$

$|P_0P_1| = |P_1P_0|$, so $\lim\limits_{\Delta\varphi\to 0}\frac{2|P_1P_0\times P_1P_2|}{|P_0P_2||P_1P_0||P_1P_2|} = \lim\limits_{\Delta\varphi\to 0}\frac{2|P_1P_0\times P_1P_2|}{|P_0P_2||P_0P_1||P_1P_2|}$, and the approximate curvature of the point P_1 can be estimated as:

$$k_{p_1} \approx \frac{2\left|P_1P_0 \times P_1P_2\right|}{\left|P_0P_1\right|\left|P_0P_2\right|\left|P_1P_2\right|}$$ (4)
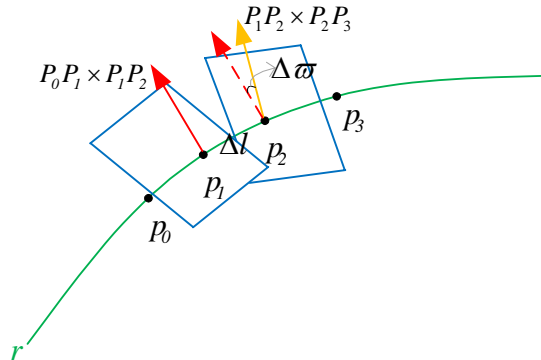

Fig. 2 *Schematic diagram of torsion estimation*

The torsion of point P_2 represents the deviation between the arc P_1P_2 and the osculating planes $\Delta P_0P_1P_2$ and $\Delta P_1P_2P_3$. Let $\Delta\varpi$ be the angle between the normal vectors of the two osculating planes, that is, the angle between the vectors $P_0P_1 \times P_1P_2$ and $P_1P_2 \times P_2P_3$, as shown in Figure 2. Then:

$$
\begin{aligned}
\tau_{p_2} &= \lim_{\Delta l = l(P_1,P_2)\to 0}\frac{\Delta\varpi}{\Delta l} = \pm\lim_{\Delta s\to 0}\frac{sin(\Delta\varpi)}{|P_1P_2|}\frac{|P_1P_2|}{\Delta l} \\
&= \pm\lim_{\Delta l\to 0}\frac{sin(\Delta\varpi)}{|P_1P_2|} = \pm\lim_{\Delta\varpi\to 0}\frac{\left|(P_0P_1\times P_1P_2)\times(P_1P_2\times P_2P_3)\right|}{\left|P_0P_1\times P_1P_2\right|\left|P_1P_2\times P_2P_3\right|\left|P_1P_2\right|} \\
&= \pm\lim_{\Delta\varpi\to 0}\frac{\left|\left((P_0P_1\times P_1P_2)\cdot P_2P_3\right)\right|\left|P_1P_2\right|}{\left|P_0P_1\times P_1P_2\right|\left|P_1P_2\times P_2P_3\right|\left|P_1P_2\right|} \\
&= \lim_{\Delta\varpi\to 0}\frac{\left((P_0P_1\times P_1P_2)\cdot P_2P_3\right)}{\left|P_0P_1\times P_1P_2\right|\left|P_1P_2\times P_2P_3\right|}
\end{aligned}
$$ (5)

The torsion of point P_2 can be expressed as:

$$\tau_{p_2} \approx \frac{\left(P_0P_1\times P_1P_2\right)\bullet P_2P_3}{\left|P_0P_1\times P_1P_2\right|\left|P_1P_2\times P_2P_3\right|}$$ (6)

The above calculation method is similar to the idea proposed in the literature [16]. The main idea is to make full use of multiple points in the vicinity to calculate the curvature and torsion, so as to eliminate noise interference and lay a solid foundation for the subsequent matching.

3. CURVE CONTOUR MATCHING ALGORITHM BASED ON DDTW

Accurate fragment reassembly requires not only a powerful shape description method but also an efficient shape matching method. The simplest matching method is to compare all the elements of two strings one by one and find the longest common substring (LCS) [17], but it is not so applicable when fragments have partial edge features missing. Another method is to look for the longest common subsequence (TLCS) [18-19]. This method does not require

continuous matching and allows for the addition, deletion and local differences of some elements in the feature string, but the precondition is that the ratio between the two arc lengths compared is equivalent to that between the actual lengths and that the sampling points should be distributed as evenly as possible. The geometric information of the 3D fragments is acquired by the 3D scanner. The density of the data points may not be even due to the different scanning resolutions and angles. In practical applications, due to various factors, such as defects or wears of the fragments, the contours of the two fragments do not exactly match each other (only some of the line segments fit each other), as shown in Figure 3. This means that LCS and TLCS algorithms are not quite effective when directly used.
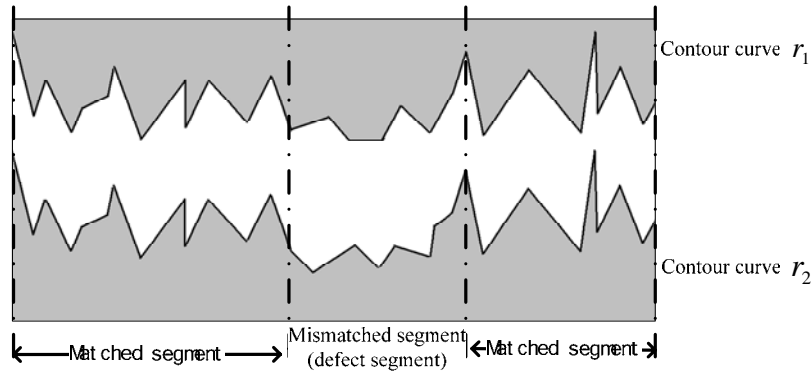

Fig. 3 *Matching relationship between defective fragments*

Dynamic time warping (DTW) [20] is one of the classical algorithms for pattern recognition. Being able to well solve the matching between two sequences with different time scales, it is widely used in speech recognition, signature identification, gesture recognition, and data mining, etc. However, for data points with the same amplitude but different trends in two sequences, the DTW algorithm tends to have mismatches, resulting in an unsatisfactory pairing effect. In recent years, some scholars have proposed derivative dynamic time warping (DDTW). Compared with DTW, DDTW not only takes into account the amplitude of data points but also more reasonably adjusts the time sequence according to the first derivative values at the data points. Therefore, this paper adopts the improved DDTW algorithm to solve the matching of contour curves.

Let the distributions of the contour curves to be compared, be $r_1 = \{p_1, p_2, \cdots, p_m\}$ and $r_2 = \{q_1, q_2, \cdots, q_n\}$. Calculate the curvature and torsion of each point using the equations (3) and (4) respectively. The shape information of the two curves can be represented by sequence A and sequence B.

$$\begin{cases} A = \{[k_{P_1}, \tau_{P_1}], [k_{P_2}, \tau_{P_2}], [k_{P_3}, \tau_{P_3}], \cdots, [k_{P_{m-1}}, \tau_{P_{m-1}}], [k_{P_m}, \tau_{P_m}]\} \\ B = \{[k_{q_1}, \tau_{q_1}], [k_{q_2}, \tau_{q_2}], [k_{q_3}, \tau_{q_3}], \cdots, [k_{q_{n-1}}, \tau_{q_{n-1}}], [k_{q_n}, \tau_{q_n}]\} \end{cases} \tag{7}$$

Let the shape information derivative at any point p_i in sequence A be expressed as:

$$\begin{cases} d_{k_i}^A = \dfrac{(k_{p_i} - k_{p_{i-1}}) + (k_{p_{i+1}} - k_{p_{i-1}})/2}{2} \\ d_{\tau_i}^A = \dfrac{(\tau_{p_i} - \tau_{p_{i-1}}) + (\tau_{p_{i+1}} - \tau_{p_{i-1}})/2}{2} \end{cases}, \quad 2 \le i \le n-1 \tag{8}$$

In equation (10), let $\begin{cases} d_{k_1}^A = d_{k_2}^A \\ d_{\tau_1}^A = d_{\tau_2}^A \end{cases}$ and $\begin{cases} d_{k_m}^A = d_{k_{m-1}}^A \\ d_{\tau_m}^A = d_{\tau_{m-1}}^A \end{cases}$. Accordingly, the shape information derivatives $d_{k_i}^B$ and $d_{\tau_i}^B$ of the sequence B at q_j can also be calculated with equation (2). Then the matching costs of any two points p_i and q_j can be expressed as:

$$d(p_i, q_j) = \sqrt{\lambda(d_{k_i}^A - d_{k_j}^B)^2 + (1-\lambda)(d_{\tau_i}^A - d_{\tau_j}^B)^2} \tag{9}$$

where, $\lambda = 0.5$, and without loss of generality, $m \leq n$. Then, based on the matching errors between all the points on the two contour curves r_1 and r_2, the matching cost matrix D between sequence A and sequence B can be expressed as:

$$D = \begin{bmatrix} d(p_1, q_1) & d(p_1, q_2) & \cdots & d(p_1, q_n) \\ d(p_2, q_1) & d(p_2, q_2) & \cdots & d(p_2, q_n) \\ \vdots & \vdots & d(p_i, q_j) & \vdots \\ d(p_m, q_1) & d(p_m, q_2) & \cdots & d(p_m, q_n) \end{bmatrix} \tag{10}$$

To solve the matching problem of sequence A and sequence B, it is necessary to find a reasonable contour point matching method in matrix D, so as to minimize the global matching cost of sequences A and B. The cumulative distance $f_{AB}(i,j)$ represents the sum of the matching costs incurred by all nodes on the shortest path from the starting point to the point (i,j) in the matrix D, and it can be expressed as:

$$f_{AB}(i,j) = d(p_i, q_j) + min(f_{AB}(i-1, j-1), f_{AB}(i-1, j), f_{AB}(i, j-1)) \tag{11}$$

The value of $f_{AB}(m,n)$ calculated according to the above equation is the minimum global matching cost of the two curves. The smaller the value, the smaller the matching distance between sequence A and sequence B, that is, the more similar the contour curves r_1 and r_2.

4. SCREENING OF THE OPTIMAL MATCHING CURVE

Thin-wall fragments may have defective edges, especially after the erosion by wind, water, and other chemicals, so in actual practice, the matching cost of two adjacent contour curves may not be necessarily the smallest. On the other hand, there may be multiple points with similar curvatures and torsions. For this reason, we need to further verify the candidate matching curves (Algorithm 1).

The DDTW algorithm is used to find the n curves r_i with the least costs of matching with the contour curve r_0, where $i = \{1, 2, \cdots, n\}$ and the larger the n value is, the larger the calculation amount of the algorithm will be. In this paper, $n = 3$, i.e. there are 3 candidate matching curves in total. For the curve r_0 and any candidate matching curve r_i, m points are sampled at equal intervals in an orderly manner from the starting point of matching, that is, $r_0 \succ \{p_1^0, p_2^0, \cdots, p_m^0\}$ and $r_i \succ \{p_1^i, p_2^i, \cdots, p_m^i\}$, where $\succ$ represents sampling. The Principal Component Analysis (PCA) [21] is used to calculate the normal vectors of each point $r_0 : \{n_1^0, n_2^0, \cdots, n_m^0\}$ and $r_i : \{n_1^i, n_2^i, \cdots, n_m^i\}$. Then voting is conducted on the consistency of the normal vector angle between the matching points to further verify the matching of curves r_0 and r_i. The specific method is as follows: set the angle threshold Δ_θ, $\theta_0 = \angle(n_j^0, n_{j+1}^0)$ and $\theta_i = \angle(n_j^i, n_{j+1}^i)$, where $i = \{1, 2, \cdots, n\}$ and $j = \{1, 2, \cdots, m\}$. If $|\theta_0 - \theta_i| < \Delta_\theta$, the number of votes for the curve r_i is increased by 1. Traverse all points in turn. Finally, the optimal matching curve for the curve r_0 is determined based on the number of votes.

Algorithm 1 Screen out the best matching curve with curve r_0

Input: the n curves r_i with the least costs of matching with the contour curve r_0, In this paper, $n = 3$.

Output: The sequence number (S)of curve r_i' that best matches curve r_0.

Begin
  int best_match_curve (){
    int $k[n+1] = 0$;
    int $S = 0$;
    for ($i = 1$; $i \leq n$; $i++$){
        1: For the curve r_0 and any candidate matching curve r_i, m points are sampled at equal intervals in an orderly manner from the starting point of matching, that is, $r_0 \succ \{p_1^0, p_2^0, \cdots, p_m^0\}$ and $r_i \succ \{p_1^i, p_2^i, \cdots, p_m^i\}$, where $\succ$ represents sampling.
        2: Calculate the normal vectors of each point r_0: $\{n_1^0, n_2^0, \cdots, n_m^0\}$ and r_i: $\{n_1^i, n_2^i, \cdots, n_m^i\}$
        3: *for* ($j = 1$; $j \leq m$; $j++$){
            Calculate the angle difference of normal vector $\theta_i = \angle(n_j^i, n_{j+1}^i)$ and $\theta_0 = \angle(n_j^0, n_{j+1}^0)$;
            If ($|\theta_0 - \theta_i| < \Delta_\theta$) $k[i] = k[i]+1$ }}
    for ($i = 1$; $i \leq n$; $i++$){
        if(k[i] > k[s]) s = i;}
    return S ;}
End

---

5. FRAGMENT REASSEMBLY

In summary, according to the determined matching curves, the matching pair of fragments F_i and F_j can be determined. Since the positions and directions of the two fragments are different, it is necessary to transform the fragment file data according to the correspondence relationships of the contour curve matching points, that is, obtaining the transformation vectors R and T of the rigid body.

Suppose the optimal matching curve of the curve r_i is r_j, and that the points of the curve r_j: $\{p_1^j, p_2^j, \cdots, p_m^j\}$ are obtained by rotating and translating the points on the curve r_i: $\{p_1^i, p_2^i, \cdots, p_m^i\}$, as shown in the following equation.

$$p_k^j = R p_k^i + T \tag{13}$$

Given the coordinates of each point on the matching curve, R and T can be estimated using the least square method to further minimize the equation (14). The iterative calculation is carried out using the method in literature [22] to gradually approximate the exact value.

$$E = \sum_{k=1}^{m} \left\| p_k^j - (R p_k^i + T) \right\| \tag{14}$$

With the R and T transformed fragment files, the adjacent fragments are finally aligned precisely by the Iterative Closest Point (ICP) method.

6. EXPERIMENT AND RESULTS ANALYSIS

The algorithm was implemented in the *C++* language on the VS2010 platform, and the OpenGL function was invoked to display the point cloud. It was applied to the reassembly of ceramic fragments. The point cloud data of the fragments were obtained by non-professionals using the Handyscan 3D handheld scanner, with a resolution of *0.5 mm*. The experimental data retrieved were not denoised, and the contour curves of the models were obtained by the method based on curvature extreme points [23]. The specific fragment models and contour edge curves are shown in Figure 4.
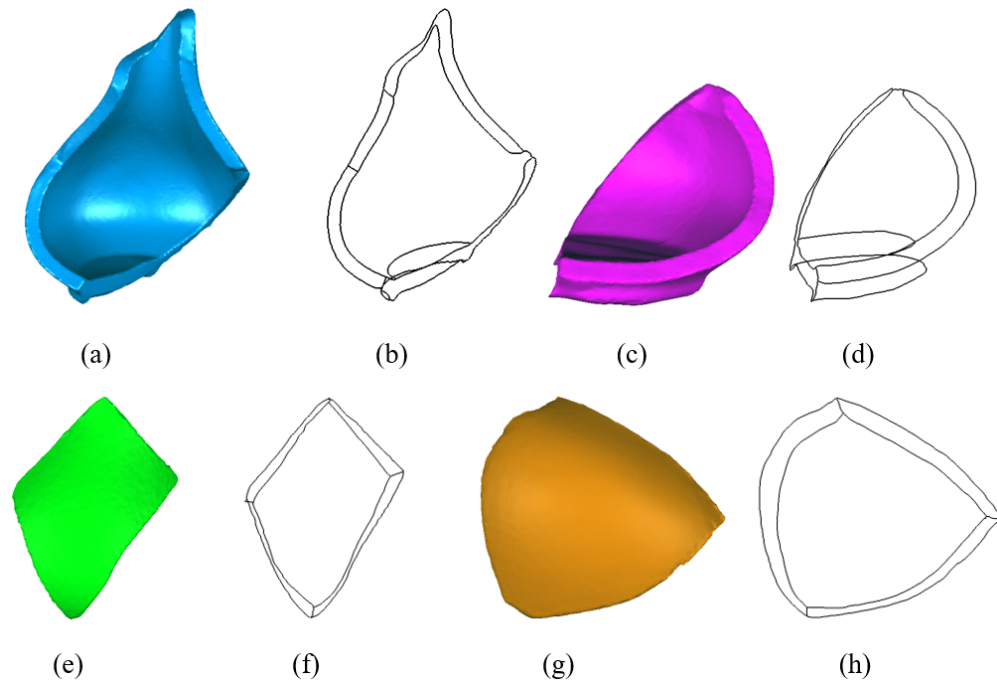

(a)	(b)	(c)	(d)
(e)	(f)	(g)	(h)

Fig. 4 *Fragment models and corresponding contour curves; (a) fragment 1; (b) contour curve of fragment 1;(c) fragment 2; (d) contour curve of fragment 2; (e) fragment 3 (f) contour curve of fragment 3; (g) fragment 4; (h) contour curve of fragment 4*

Since a ceramic fragment has a certain thickness, the contour curve is composed of two similar curves, as shown in Figure 4 (b) and (d). Firstly, the authors used the method in reference [24] to separate the contour curve from the intersections into several separate curves, then matched the contour curves of fragment 1 and fragment 2 using the DDTW method to find the optimal matching line segment, and finally calculated the rotation matrix and the translation matrix according to the matching relationships of the points on the optimal matching line segment. The specific reassembly results are shown in Figure 5.
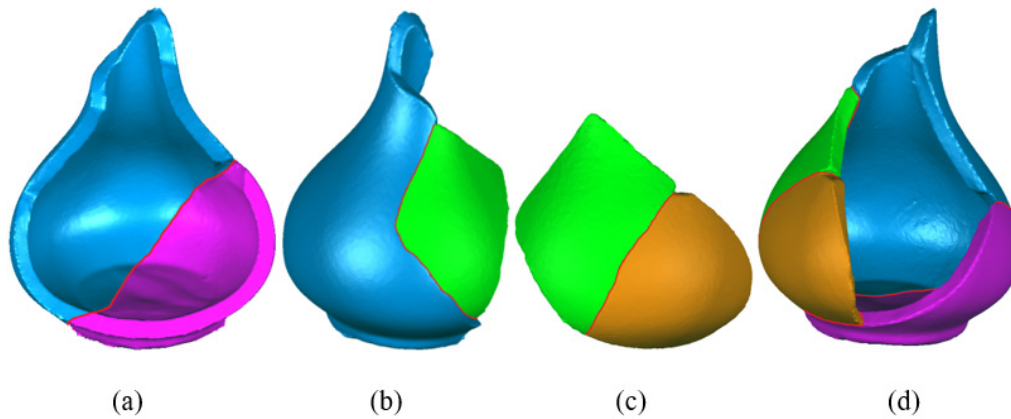
Fig. 5. *Reassembly results of different fragments; (a) reassembly of fragments 1 and 2; (b) reassembly of fragments 1 and 3; (c) reassembly of fragments 3 and 4; (d) reassembly of fragments 1 and 4*

In order to verify the performance of the algorithm, the author compared it with the DTW-based and the LCS-based algorithms and recorded the time and the errors of the reassembly. Suppose the reassembly error is δ, which refers to the average of the sum of the Euclidean distances between the pair of points on adjacent contour curves. The specific results are shown in Table 1, where t_1 and δ_1 represent the time consumed by the DTW algorithm and its reassembly error, and t_2 and δ_2 are the results of the LCS algorithm, and t_3 and δ_3 the results of this algorithm. Since the DDTW algorithm needed to perform the derivation, its running time was longer than those the DTW and the LCS algorithms. Since, in this paper, the DDTW algorithm is allowed for non-matching points, the candidate matching curves were further filtered to ensure matching, so the reassembly error of the DDTW algorithm was much smaller than those of the DTW and the LCS algorithms [25].

Table1 *Comparison of the algorithms in running time and reassembly error*

serial number of spliced fragments	time/s			reassembly error /mm		
	t_1	t_2	t_3	δ_1	δ_2	δ_3
fragments 1 and 2	15.972	14.893	17.324	1.532	1.987	0.879
fragments 1 and 3	14.672	13.267	15.213	1.345	1.766	0.794
fragments 3 and 4	12.344	12.123	12.543	1.732	1.923	0.765
fragments 2 and 4	9.323	9.212	9.892	1.268	1.564	0.532

7. CONCLUSION

In order to address the automatic reassembling of 3D thin-wall fragments, this paper proposes an automatic reassembly method of 3D thin-wall fragments based on the DDTW algorithm. Firstly, this paper designs a calculation method for discrete curvature and torsion to solve the difficulty in the calculation of curvature and torsion with discrete data points. The calculation method fully utilizes the information of multiple points in the vicinity, which can effectively eliminate the noise interferences and lay a solid foundation for the subsequent matching. Then according to the feature-invariant descriptors, i.e. curvature and torsion, this paper further searches the candidate matching line segments by the DDTW method and records the positions of the starting and ending points of each candidate matching segment. By adding the geometric invariant constraint information, it further searches the optimal matching line

segments based on the voting mechanism and eliminates mismatching. Then it adopts the least squares method to estimate the rotation and transformation matrices and uses the ICP method to complete the reassembly of fragments. At last, the feasibility of the proposed algorithm is verified by the experiment of ceramic fragments splicing; according to the analysis of the experiment results, the time-consuming of the proposed algorithm is almost the same as that of the algorithm based on DTW or LCS, but the error of the algorithm is only 1/2 of DTW and 2/5 of LCS, which shows that the algorithm has higher precision. In addition, although the DDTW algorithm allows non-matching points, if the edge contour of the fragment is damaged too seriously, the proposed algorithm cannot extract matching line segments, which eventually leads to the error of fragment merging.

8.   ACKNOWLEDGMENTS

9.   REFERENCES

[1]   K. Zhang, W. Yu, M. Manhein, W. Waggenspack, 3D Fragment Reassembly Using Integrated Template Guidance and Fracture-Region Matching, *IEEE International Conference on Computer Vision, IEEE Computer Society*, pp. 2138-2146, 2015.

https://doi.org/10.1109/ICCV.2015.247

[2]   G.S. Xu, Y.B. Zhang, Y. Lin, Y.J. Lin, Research on Fragments Reassembly Based on Feature of Chinese Character and Template Matching, *Computer & Information Science*, Vol. 7, No. 3, pp. 92-92, 2014. https://doi.org/10.5539/cis.v7n3p92

[3]   H.J. Wolfson, On curve matching, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Vol. 12, No. 5, pp. 483-489, 1990. https://doi.org/10.1109/34.55108

[4]   H.C.D.G. Leitão, J. Stolfi. A Multiscale Method for the Reassembly of Fragmented Objects, *British Machine Vision Conference*, Vol. 24, No. 9, pp. 1239-1251, 2008.

https://doi.org/10.1109/TPAMI.2002.1033215

[5]   G. Üçoluk, I.H. Toroslu, Automatic reconstruction of broken 3-D surface objects, *Computers & Graphics*, Vol. 23, No. 4, pp. 573-582, 1999.

https://doi.org/10.1016/S0097-8493(99)00075-8

[6]   G. Oxholm, K. Nishino, Reassembling thin artifacts of unknown geometry, International *Conference on Virtual Reality, Archaeology and Cultural Heritage, Eurographics Association*, pp. 49-56, 2011.

[7]   W. Kong, B.B. Kimia, On solving 2D and 3D puzzles using curve matching, *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Vol. 2, pp. 583-590, 2001.

[8]   Y.H. Zhang, G.H. Geng, X.R. Wei, J. Zhang, Reassembly of Fractured Fragments Based on Skeleton Graphs Matching, *Zidonghua Xuebao/acta Automatica Sinica*, Vol. 43, No. 4, pp. 622-633, 2017.

[9]   S. Li, G. Geng, M. Zhou, J. Li, Interactive reassembly of fractured fragments based on surface adjacency constraint, *Journal of Computer-Aided Design & Computer Graphics*, 2016.

[10]   M. Q. Zhou, J. Yuan, G.H. Geng, Y.H. Zhang, Interactive reassembly of fractured fragments based on feature points of contour line, *Optics and Precision Engineering*, Vol. 25, No. 6, pp. 1597-1606, 2017. https://doi.org/10.3788/OPE.20172506.1597

[11]   J.J. Li, G.H. Geng, M.Q. Zhou, S.S. Li, A computer-assisted system for virtual relic assembling based on adjacency constraint, *Journal of Northwest University*, 2016.

[12]   S. Rusinkiewicz, Efficient variant of the ICP algorithm, *Proc Ofdim*, 2001.

[13]   G. Üçoluk, I.H. Toroslu, Automatic reconstruction of broken 3-D surface objects, *Computers & Graphics*, Vol. 23, No. 4, pp. 573-582, 1999.

        https://doi.org/10.1016/S0097-8493(99)00075-8

[14]   W. Kong, B.B. Kimia, On solving 2D and 3D puzzles using curve matching, *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. IEEE*, Vol. 2, pp. 583-590, 2001.

[15]   L. Fang, W. Lu, W. Huang, Estimate algorithms and embedded crafts of curvature and torsion, *Journal of Graphics*, Vol. 33, No. 2, pp. 9-13, 2012.

[16]   J. Gao, C.M. Zhang, X.X. Meng, Z.Q. Feng, Automatic Fragment Re-Assembly Method Based on DDTW Match, *Chinese Journal of Computers*, Vol. 32, No. 2, pp. 342-349, 2009.

        https://doi.org/10.3724/SP.J.1016.2009.00342

[17]   T.A. Starikovskaya, H.W. Vildhøj, Time-Space Trade-Offs for the Longest Common Substring Problem, *CPM*, pp. 223-234, 2013.

        https://doi.org/10.1007/978-3-642-38905-4_22

[18]   A. Apostolico, C. Guerra, The longest common subsequence problem revisited, *Algorithmica*, Vol. 2, No. 1-4, pp. 315-336, 1987. https://doi.org/10.1007/BF01840365

[19]   B. Bukh, R. Hogenson, Length of the longest common subsequence between overlapping words, *arXiv:1803.03238*, pp. 1-10, 2018.

[20]   T.M. Rath, R. Manmatha, Word Image Matching Using Dynamic Time Warping, *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on. IEEE*, Vol.2, pp. 521-527, 2003.

[21]   E.J. Keogh, M.J. Pazzani, Derivative Dynamic Time Warping, 2001.

        https://doi.org/10.1137/1.9781611972719.1

[22]   S. Wold, K. Esbensen, P. Geladi, Principal component analysis, *Chemometrics & Intelligent Laboratory Systems*, Vol. 2, No. 1, pp. 37-52, 1987.

        https://doi.org/10.1016/0169-7439(87)80084-9

[23]   S.Y Fu, L.S Wu, Feature Line Extraction from Point Clouds Based on Geometric Structure of Point Space. *3D Research*, Vol. 10, No. 2, pp. 16-26, 2019.

        https://doi.org/10.1007/s13319-019-0227-x

[24]   S. Rusinkiewicz, M. Levoy, Efficient variant of the ICP algorithm, *Proc Ofdim*, Vol. 1, pp. 145-145, 2001.

[25]   H. Wei, L. Li, Curve description and matching using arch sequence, *Journal of Image & Graphics*, 2017.