



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학박사학위논문

**Reassembly of broken objects  
using surface signatures**

표면 특성을 이용한  
부러진 물체의 재조립 알고리즘

2017년 2월

서울대학교 대학원  
기계항공공학부  
손태근

## **ABSTRACT**

# **Reassembly of broken objects using surface signatures**

Tae-geun Son

School of Mechanical and Aerospace Engineering

The Graduate School

Seoul National University

As 3D object acquisition techniques and computational powers have been improved, research into the virtual reassembly of broken 3D objects has been actively conducted and techniques have been applied to various fields. However, the complicated and uneven shape of the fractured surfaces of broken objects makes it difficult to extract features for reassembling the object and forces us to use high-dimensional complex shape information. In addition, deformation of the fractured surface due to the small debris particles generated when the object is broken is one of the difficulties in this field.

In this paper, we propose a surface signature descriptor that can easily and effectively represent the complicated fracture surface of a broken object. This descriptor is inspired by the human reassembly mechanism, which matches a convex region of a fractured surface to a concave region with a similar shape of

another fractured surface. This descriptor is not dependent on the local geometry of fractured surfaces, so it has the advantage of being insensitive to the small noise that may exist in the fractured surface. In addition, it is simple and easy to calculate compared with the existing high-dimensional shape information.

In the process of reassembling the actual broken object using the proposed surface signature descriptor, we propose and improve the segmentation, calculation of surface roughness and similarity measures, definition of the surface roughness, and similarity measurement technique between the fractured surfaces so that the complex fractured surface can be processed. The effectiveness of the proposed method is demonstrated through the realization of reassembly using actual broken objects.

In addition, we propose a verification method through the creation of hypothetical fractured surfaces as a solution to the problem of the difficulty of the quantitative evaluation of reassembly technology due to the absence of verification data before a break. The virtual fractured surface is arbitrarily generated by several factors, and it can be evaluated in various aspects, such as accuracy, efficiency, and robustness against noise of the reassembly algorithm. The reassembly method proposed in this study is also evaluated.

This study is meaningful, in that it developed and improved the techniques suitable for reassembly from the beginning of the reassembly, by defining a descriptor of the 3D fractured surface, to the qualitative evaluation of the methods. It could also be expanded to other research fields related with reassembly.

**Keywords: broken object reassembly, fractured surface, surface signature,  
virtual fractured surface, reassembly assessment**

**Student Number: 2006-20941**

# CONTENTS

ABSTRACT.....	i
CONTENTS.....	iv
LIST OF TABLES.....	vi
LIST OF FIGURES.....	vii
CHAPTER 1. INTRODUCTION .....	1
1.1 Background .....	1
1.2 Research objectives .....	2
CHAPTER 2. RELATED WORKS.....	3
2.1 2D reassembly .....	3
2.2 3D reassembly .....	4
2.3 Assessment .....	6
CHAPTER 3. SURFACE SIGNATURES.....	8
3.1 Concepts.....	8
3.2 Surface signature descriptor .....	10
3.2.1 Convex/concave analysis .....	10
3.2.2 Convex/concave analysis of fractured surface .....	15
3.3 Results and discussion.....	20
CHAPTER 4. REASSEMBLING .....	25
4.1 Overview .....	25
4.2 Fractured surface extraction .....	27

4.2.1 Smoothing .....	27
4.2.2 Roughness .....	31
4.2.3 Segmentation.....	36
4.2.4 Fractured surface clustering .....	38
4.3 Fractured surface matching .....	40
4.3.1 Feature curve.....	40
4.3.2 Matching .....	43
CHAPTER 5. EXPERIMENTS .....	50
CHAPTER 6. ASSESSMENT .....	60
6.1 Assessment with virtual fractured surfaces .....	60
6.1.1 Randomly fractured surfaces.....	60
6.1.2 Reassembly with randomly fractured surfaces.....	63
6.1.3 Assessment of reassembly .....	64
6.2 Assessment with real broken objects.....	73
6.3 Discussion .....	75
CHAPTER 7. CONCLUSION.....	76
REFERENCES.....	78
ABSTRACT (Korean).....	88

## LIST OF TABLES

Table 5.1 Numerical results of pairwise matching. ....	54
Table 6.1 Reassembly experiment results with randomly generated fractured surfaces.....	69
Table 6.2 Reassembly results with varying Gaussian noise.....	72

# LIST OF FIGURES

Fig. 2.1 Multi-scale features of fractured surface and its representation. ....	5
Fig. 2.2 Virtual fracturing method using cutting objects in cultural heritage.....	7
Fig. 3.1 Representative information used in broken object matching.....	9
Fig. 3.2 Segmentation results of representative models.....	10
Fig. 3.3 Dihedral angles .....	11
Fig. 3.4 Normal deviations in 2D convex and concave vertex.....	12
Fig. 3.5 Complicated and noisy fractured surfaces.....	13
Fig. 3.6 Convex/concave analysis with one-ring neighbor. ....	14
Fig. 3.7 Concept of surface signature.....	16
Fig. 3.8 Spherical support of vertex on fractured surface .....	17
Fig. 3.9 Descriptor normal of convex vertex.....	19
Fig. 3.10 Descriptor normal of concave vertex .....	19
Fig. 3.11 Comparison of convex/concave analysis .....	21
Fig. 3.12 Surface signature examples and overlapped surface signatures .....	21
Fig. 3.13 Second surface signature examples and overlapped surface signatures ..	22
Fig. 3.14 Examples of 1:N matching case.....	24
Fig. 3.15 Overlapped surface signatures in 1:N matching case .....	24
Fig. 4.1 The overview of our reassembling process.....	26
Fig. 4.2 Examples of mesh smoothing .....	30
Fig. 4.3 Overview of Lavoué's roughness calculation algorithm.....	32

Fig. 4.4 Local windows on vertex $v$ and curvature interpolation. ....	33
Fig. 4.5 Vertex roughness as curvature difference .....	35
Fig. 4.6 Mesh segmentation and mean curvatures .....	36
Fig. 4.7 K-means clustering algorithm with $k=3$ .....	39
Fig. 4.8 Results of segment clustering .....	39
Fig. 4.9 Douglas-Peucker curve simplification and its results. ....	42
Fig. 4.10 Oriented point basis for spin image .....	44
Fig. 4.11 Curve distance metric.....	48
Fig. 5.1 Broken brick models. ....	51
Fig. 5.2 Broken cake models.....	52
Fig. 5.3 Broken sculpture models.....	53
Fig. 5.4 Pairwise matching for brick fragments .....	55
Fig. 5.5 Pairwise matching for cake fragments .....	56
Fig. 5.6 Pairwise matching for sculpture fragments.....	57
Fig. 5.7 Global reassembling results .....	59
Fig. 6.1 Effects of number of deformation seed points .....	61
Fig. 6.2 Examples of randomly fractured surfaces.....	62
Fig. 6.3 Simple reassembly examples with randomly fractured surfaces. ....	63
Fig. 6.4 Reassembly process with randomly fractured surfaces .....	64
Fig. 6.5 Fractured surfaces after reassembly. ....	65
Fig. 6.6 Inner fragment.....	66
Fig. 6.7 Outer point clouds after reassembly of box example.....	66

Fig. 6.8 Mesh deviation of reassembled box objects. ....	67
Fig. 6.9 Histogram of mesh deviation of box examples.....	68
Fig. 6.10 Effects of sigma value for Gaussian noise on the fractured surface. ....	70
Fig. 6.11 Difference between original model and reassembled model.....	71
Fig. 6.12 Calculation times for reassembly with varying Gaussian noise.....	71
Fig. 6.14 Misaligned fragments with strong Gaussian noise. ....	72
Fig. 6.14 Original model and fragments of gypsum mask model. ....	73
Fig. 6.15 Reassembled mask fragments and its outer point clouds.....	74
Fig. 6.16 Assessment of mask model reassembly. ....	75

# CHAPTER 1.

## INTRODUCTION

### 1.1 Background

With the improvement of computing power, many jobs that previously relied on people's cognitive abilities are now performed by computers. Many tasks, such as recognition, retrieval, and matching, are automated using 2D and 3D data, and the development of 3D object acquisition technology has further accelerated this phenomenon. In response to this trend, the solving of 2D and 3D puzzles stimulated people's curiosity and increased interest in the problem of the reassembly of broken 3D objects.

The reassembly of broken objects has long been an issue in archeology. Recently, 3D object reassembly technology has been extended to the medical field, and it has been used as a simulation and planning technique for restoring broken bones in plastic surgery and orthopedic surgery. However, reassembly technology, which can be used in various ways, poses several difficulties.

The first is the complex shape of the fractured surface. The fractured surface generated by the fracture of an object is the result of the combination of the structural and chemical characteristics of the object and the external environmental factors at the time of the fracture. Therefore, it is very difficult to define the rules or features, and as a result, fractures mostly have complex and uneven shapes. In addition, the shape deformation of the fracture surface caused by the separation of

the small fragments in the fracture surface makes the matching between the fracture surfaces more difficult.

Secondly, there is the lack of information that can be utilized when the fragment-matching process is needed. A single fragment is able to combine with a certain number of other fragments, and it is difficult to define the relation of matching between fragments. Furthermore, whether matching between fractured surfaces in fragments is one-on-one or partial during virtual reassembly is uncertain.

Finally, there is no information about the shape before the fracture, and it is not easy to evaluate the reassembly process quantitatively. As mentioned earlier, it is very difficult to define the fracturing mechanism of a 3D object, and even if it is defined, it must be governed by a vastly large computational complexity and parameter space. Because of this difficulty, many of the previously proposed reassembly methods have only been referred to as plausible results and their performance can be easily changed by a number of factors.

## **1.2 Research objectives**

To accelerate and improve the effectiveness of the reassembly process, we define a novel descriptor that can simply display the features of complicated and uneven fractured surfaces. Several techniques for the reassembly process, such as segmentation, definition of roughness, and estimation of the similarity between fractured surfaces, are modified so that they are appropriate for broken objects. A quantitative evaluation of the process is also designed.

## **CHAPTER 2.**

### **RELATED WORKS**

Since the introduction of the reassembly of broken objects for the first time in the late 1990s, many research studies have been conducted on the subject and applied to the fields of archeology [1–3], paleontology [4], and medicine [5, 6]. In this study, we describe the existing studies related to reassembly by dividing them into 2D and 3D based on object shape and introduce the recent assessment research.

#### **2.1 2D reassembly**

In the beginning, the reassembly of broken objects was considered the matching of break curves. These studies were inspired by 2D jigsaw puzzles [7, 8]. The information we use to align 2D jigsaw puzzles includes the shape of the puzzle piece, the texture of the puzzle piece, its association with the boundary frame, and the puzzle outline drawn on the bottom plate, if available. Some of these types of information are more important than others, but the most basic thing to consider when completing a puzzle is the consistency with the shape of the next piece of the puzzle. If the shape of the puzzle piece does not fit, the two pieces cannot be placed together. In 2D reassembly techniques, such as jigsaw puzzle techniques, the bounding curves of fragments are extracted from an almost planar shape, such as fractured tiles [9–11], and clustered by their shapes to find the most similar curve.

In the case of an axially symmetric shape, such as a vessel or piece of pottery, a complete global surface is estimated from an incomplete set of pieces [12].

## **2.2 3D reassembly**

For the reassembly of 3D solid objects, Papaionnou et al. introduced a depth buffer, a 2D array map representing the distance of the facet from a plane perpendicular to the average facet normal [2]. A depth buffer can be used to determine the fractured surface and to estimate the matching error between two surfaces. Later, boundaries of fractured surfaces were extracted and utilized to perform matching error estimation simultaneously to extend the method [13]. This method is very simple and intuitive, but it is very sensitive to the projection plane for the depth buffer, and it is hard to apply to curved fracture surfaces.

Huang et al. defined multi-scale surface characteristics based on the integral invariant descriptor (Fig. 2.1) [14]. Patch-based surface features are created by clustering multi-scale surface characteristics and compared to find pairwise matching between two fractured surfaces. In addition, constrained local registration is conducted simultaneously to achieve non-penetrating alignment of fragments. This method has the advantage of providing a large amount of information and matching possibility by representing a single fractured surface as a set of multiple overlapping features, but the efforts required to define and calculate such features are too great.

Winkelbach and Wahl proposed the pairwise matching of 3D fragments using

cluster trees [15]. In this method, a hierarchical clustering algorithm is applied to decompose each point set of fragments into a binary tree structure, and then pairwise fragment matching is conducted by descending the cluster trees simultaneously in a depth-first fashion.

Recently, Altantsetseg et al. proposed the pairwise matching of broken fragments from unorganized point clouds [16]. They introduced a new descriptor containing the cluster of feature points and curves along the principal directions of the cluster. They compared the Fourier coefficients of each curve and the total energies of the curves to find pairwise matches.

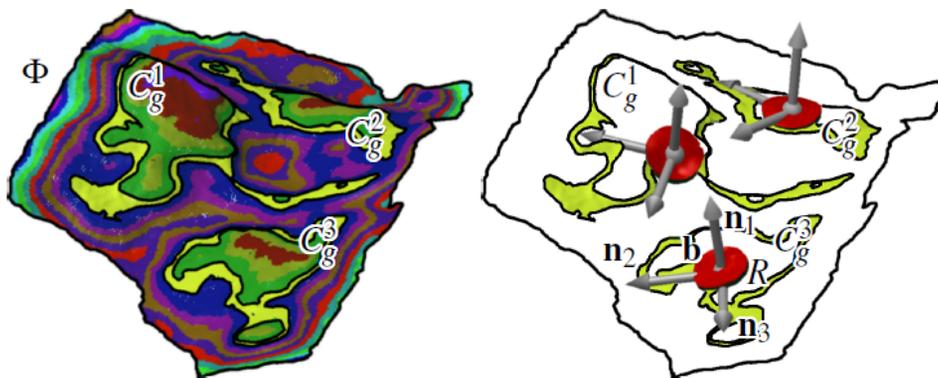


Fig. 2.1 Multi-scale features of fractured surface and its representation [14].

## 2.3 Assessment

The main reason for the difficulty of the quantitative evaluation of reassembly is the absence of ground truth. Especially in archeology, there is no information about the original shape of a broken object. Therefore, there is no standard with which to compare the results of reassembly, and only quantitative evaluation of geometric alignment results and performance is performed.

Recently, virtual fracturing technology for the evaluation of reassembly and retrieval has been studied [17]. To reproduce the fracture process, a fairly complex and plausible physical model is required. The structural and material properties of the object and the mechanical and chemical processes at the time of fracture should be defined. In the field of archeology, time-dependent forces acting outside the object should also be defined, and the long-term effects of erosion caused by the material and environmental factors of the object should also be considered. These properties are difficult to define precisely, even by experts, and even if defined, they require a vastly large computational complexity and parameter space.

Recently, however, the heuristic method has been expanded to include the movie production field as well as the physically exact model [18]. This is because, in existing reassembly and retrieval techniques, the overall characteristics of the shapes are used rather than the material or environmental condition of the objects, and the results of the fracturing techniques, which are not based on the physical model, are considered very plausible for the audience. Based on this idea, studies were conducted to define the fracture line or cutting plane on objects [19–21], and

a study was conducted to simulate the fracture of cultural heritages using irregular cutting objects (Fig. 2.1) [18].



Fig. 2.2 Virtual fracturing method using cutting objects in cultural heritage [18].

# CHAPTER 3.

## SURFACE SIGNATURES

### 3.1 Concepts

A person uses various kinds of information when fitting a puzzle or a broken object. In the case of fitting a 2D jigsaw puzzle, the shape of the puzzle piece, the entire puzzle frame (particularly at the corner), and the texture of the puzzle piece are used. When assembling a 3D object, we try to match two fractured surfaces of two broken fragments and check if they fit well, even though it is difficult to recognize the fractured surfaces immediately. Moreover, we check how smoothly continued the intact (or unbroken) surfaces adjacent to the fractured surfaces are after aligning the fractured surfaces. Here, the convex shape of one fractured surface should match with the concave shape of the opposite fractured surface, and the concave shape of one fractured surface should match with the convex shape of the opposite side (see Fig. 3.1). The surface signature was designed based on this human perception for reassembly.

Essentially, the surface signature utilizes the convex and concave information of the vertices on the fracture surface. The convex and concave information of each vertex is extracted first, and the distributions of each convex and concave region are evaluated. Then, based on this information, matching pairs of fractured surfaces are detected.

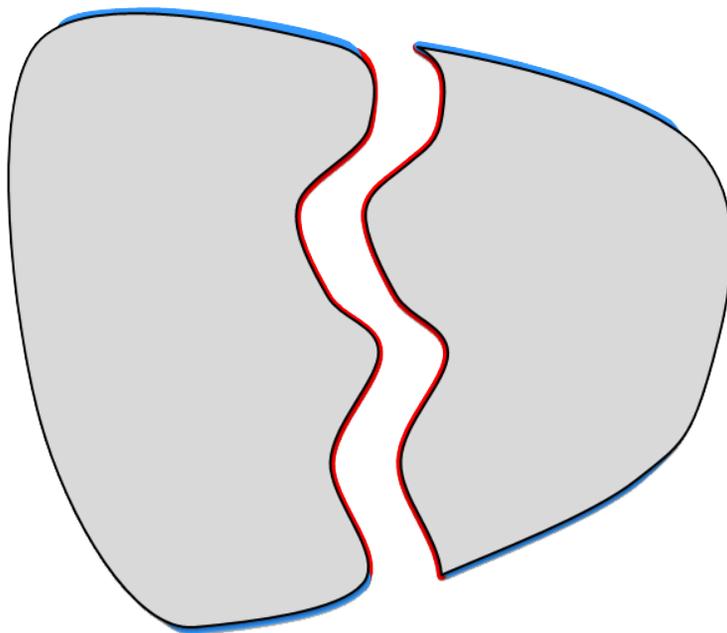


Fig. 3.1 Representative information used in broken object matching. Shape of broken part (red), smoothly continued boundaries (blue), and texture if it exists (gray).

## 3.2 Surface signature descriptor

### 3.2.1 Convex/concave analysis

Typical 3D objects are represented by a set of polygons, which consists of a set of vertices  $V = \{\mathbf{v}_i\} \subset \mathcal{R}^3$  and a set of faces  $F\{f_k\} = \{\text{polygon}(\mathbf{v}_{k1}, \dots, \mathbf{v}_{kn})\}$  made of  $n$  vertices. In this research, the polygon mesh is a triangular mesh, which is the most common in this field, and the face is represented by  $f_k = \Delta(\mathbf{v}_{k1}, \mathbf{v}_{k2}, \mathbf{v}_{k3})$ .

Convex/concave analysis is used to analyze the geometry of the shape and perform certain applications, such as segmentation [23–28]. This originates from the fact that people separate objects into meaningful components, passing the boundary of components through the deep concavity of the object [29]. Some examples of distinguishing an object by its meaningful components in this way are shown in Fig. 3.2 [28].



Fig. 3.2 Segmentation results of representative models in Princeton Segmentation Benchmark [28].

Previous studies have used two basic criteria for distinguishing a vertex as convex and concave. One of them is the dihedral angle criteria. The dihedral angle is the internal angle between two planes sharing an edge connecting two vertices, as shown in Fig. 3.3. If this angle is less than  $\pi$  (180 degrees), the edge is considered convex; otherwise, it is concave. The convex and concave parts of a vertex  $v$  on the surface mesh are determined by the convex and concave states of the edges sharing the vertex  $v$ .

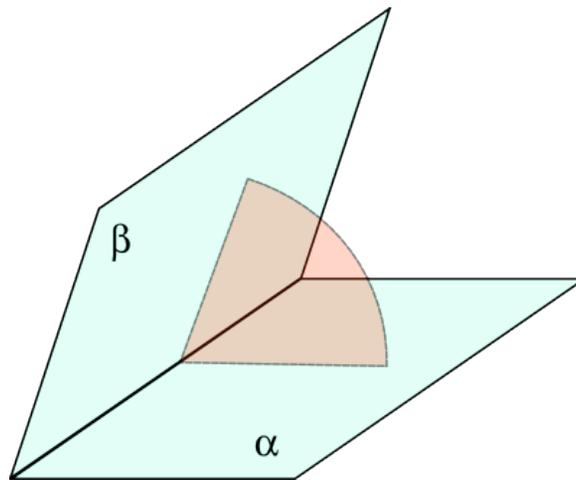


Fig. 3.3 Dihedral angles

There is also a method using the vertex normal directly instead of using dihedral angles [28]. The relation between a vertex  $\mathbf{v}$  of the surface mesh and the neighboring vertices of vertex  $\mathbf{v}$  can be expressed as follows:

$$\frac{(\mathbf{v}_i - \mathbf{v}_j)}{\|\mathbf{v}_i - \mathbf{v}_j\|} \cdot (\mathbf{n}_j - \mathbf{n}_i) > \zeta$$

where  $\mathbf{n}_i$  and  $\mathbf{n}_j$  are the vertex normal vectors of  $\mathbf{v}_i$  and  $\mathbf{v}_j$ , respectively, and  $\zeta$  is a small constant. A vertex  $\mathbf{v}_i$  is determined to be concave when there is at least one  $\mathbf{v}_j$  satisfying the above equation in the neighboring vertex. Fig. 3.4 delineates the difference of the equation in each convex and concave state.

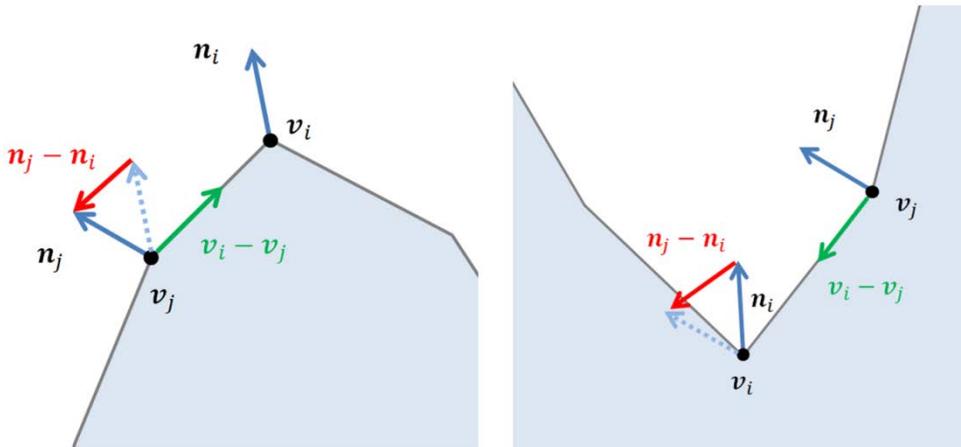


Fig. 3.4 Normal deviations in 2D convex and concave vertex

However, these convex/concave analysis methods are difficult to apply to broken objects because of the complicated and noisy characteristics of the fractured surfaces of broken objects. With the above methods, a convex/concave analysis of a vertex is usually performed based on the geometric relationship between a vertex and its neighboring vertices. However, in fractured surfaces with complicated shapes, as shown in Fig. 3.5, the geometric relations between vertices are very unstable, so adopting previous methods is difficult. Even if the analysis is conducted on a fractured surface, the results cannot represent the shape characteristics of the fractured surface appropriately.

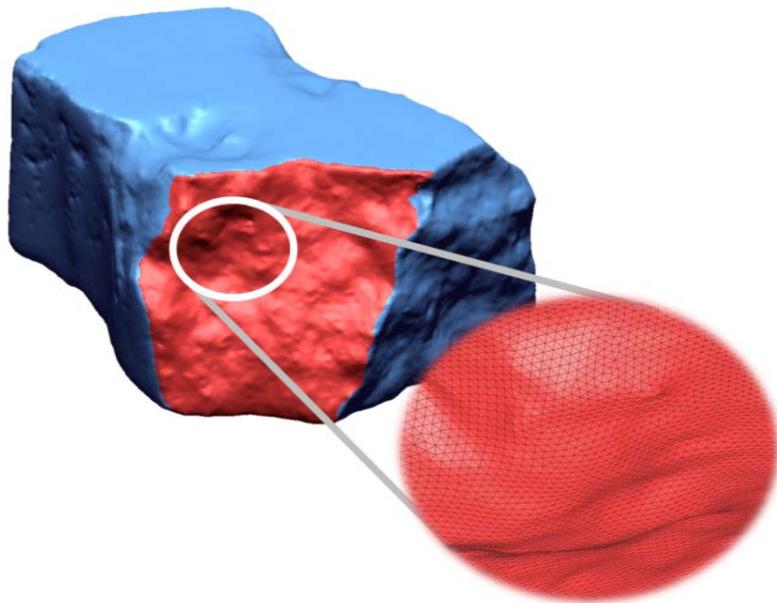


Fig. 3.5 Complicated and noisy fractured surfaces

Fig. 3.6 is an example of a convex/concave analysis based on the normal deviation through the one-ring neighbors. The convex and concave vertices distributions are rendered in red and blue, respectively. Although the two fractured surfaces have a counterpart relationship (matching pair in reassembly), it is hard to find any similarity between the convex/concave distributions of the surface. Here, an analysis or description of the surface that is robust to noise and can be applied to the fractured surfaces is required.

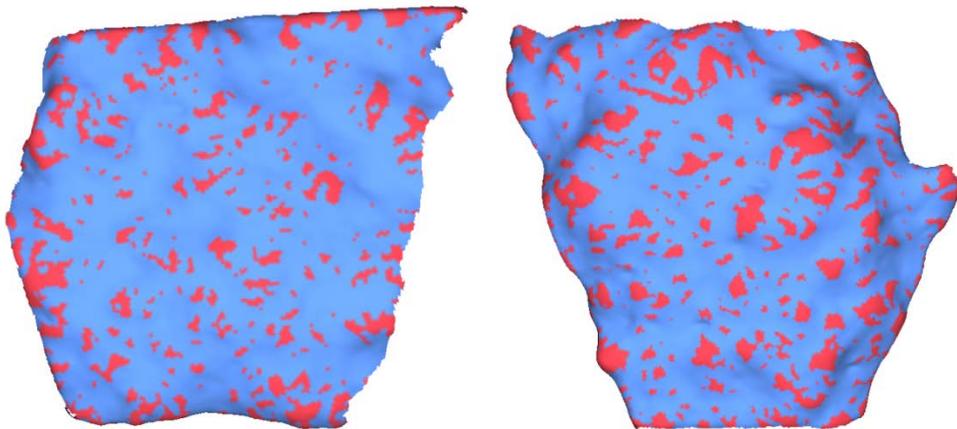


Fig. 3.6 Convex/concave analysis with one-ring neighbor.

### 3.2.2 Convex/concave analysis of fractured surface

A new convex/concave analysis of a fractured surface with a complicated and noisy shape was inspired by a local reference frame, a 3D descriptor used in shape matching [31]. The local reference frame of a vertex on the mesh is defined based on the surrounding geometry of the vertex. There are many ways to define a local reference frame [30], but it is essentially based on the eigenvector of the covariance matrix of the points on the spherical support of radius  $r$  centered at a vertex. The methods of calculating the covariance matrix and determining each axis of the local reference frame are classified into various types [31–34].

We have devised a new signature descriptor, a surface signature, which can represent the convex and concave information of a vertex on a fractured surface using the local reference frame method. The surface signature separates all points on the fractured surface into convex and concave regions, which are then used for fractured surface matching. The basic concept of the surface signature is illustrated in Fig. 3.7. The convex and concave analysis of the vertex on the fractured surface is based on the normal axis of the descriptor defined at each vertex. In Fig. 3.7, the thin arrows represent the descriptor normal computed by a given rule. The descriptor normal defined in all vertices is divided into convex and concave vertices by comparing the average normal (thick green arrow in the figure) of the whole fractured surface.

In the example shown in Fig. 3.7, the red arrows represent descriptor normals on the convex vertex, and the blue arrows represent those on the concave vertex. The

fractured surfaces divided into convex and concave regions by the surface signature are compared with other fractured surfaces based on their distribution patterns.

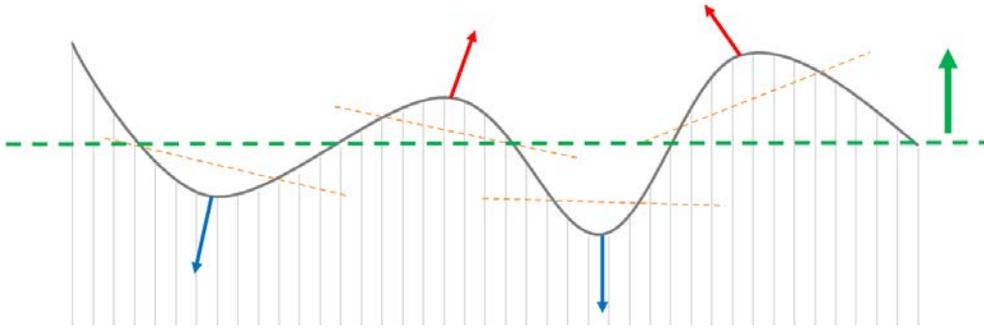


Fig. 3.7 Concept of surface signature. The surface signature can divide vertices on fractured surfaces into convex and concave regions.

The calculation of the surface signature begins with defining a spherical support with radius  $r$  on a vertex  $\mathbf{v}$  upon the fractured surface. From the spherical support, we can obtain intersection points  $\{\mathbf{p}_i\}$  between the fractured surface and the spherical support (blue dots in Fig. 3.8). Then, the fitting plane of intersection points  $\{\mathbf{p}_i\}$  that passes through the barycenter of the intersection points  $\hat{\mathbf{p}}$

$$\hat{\mathbf{p}} = \frac{1}{n} \sum_{i=0}^n \mathbf{p}_i$$

with the normal vector given by the normalized eigenvector of the covariance matrix

$$\Sigma_{\hat{\mathbf{p}}} = \frac{1}{n} \sum_{i=0}^n (\mathbf{p}_i - \hat{\mathbf{p}})(\mathbf{p}_i - \hat{\mathbf{p}})^T$$

can be derived. However, the eigenvector defines the principal direction, and the sign of the vector is not clear. Here, we propose a novel criterion for the sign decision of the descriptor normal.

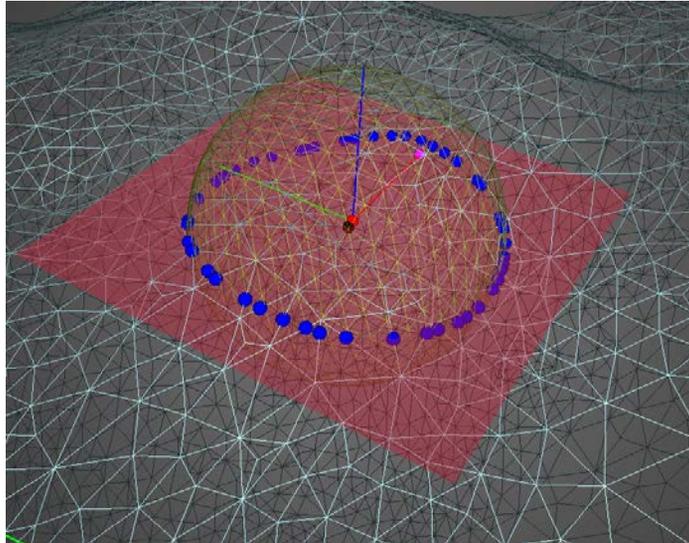


Fig. 3.8 Spherical support of vertex on fractured surface. Blue dots indicate intersecting points between fractured surface and spherical support.

The conventional definition method of the local reference frame is set to be the same as the surface normal direction of the reference point. In this study, the sign of the vector is set so that it can have a consistent direction in both convex and concave situations.

When the eigenvector having the smallest eigenvalue calculated from the covariance matrix is  $\mathbf{n}_e$  (i.e., the normal vector of the fitting plane) and the surface normal at the vertex  $\mathbf{v}$  is  $\mathbf{n}_{surf}$ , the descriptor normal  $\mathbf{n}_{desc}$  can be calculated as follows:

$$\mathbf{n}_{desc} = \begin{cases} -\mathbf{n}_e, & \mathbf{n}_e \cdot (\hat{\mathbf{p}} - \mathbf{v}) / \mathbf{n}_e \cdot \mathbf{n}_{surf} \geq 0 \\ \mathbf{n}_e, & otherwise \end{cases}$$

The equation defines the descriptor normal with a consistent direction in each convex and concave situation. Geometrically, the equation makes the descriptor normal set opposite to the fitting plane's position. Eventually, for every vertex on the fractured surface, the descriptor normal is directed out of the surface in the case of a convex situation, and it is directed inside of the surface in the case of a concave situation. Figs. 3.9 and 3.10 show the direction of the descriptor normal in convex and concave situations, respectively.

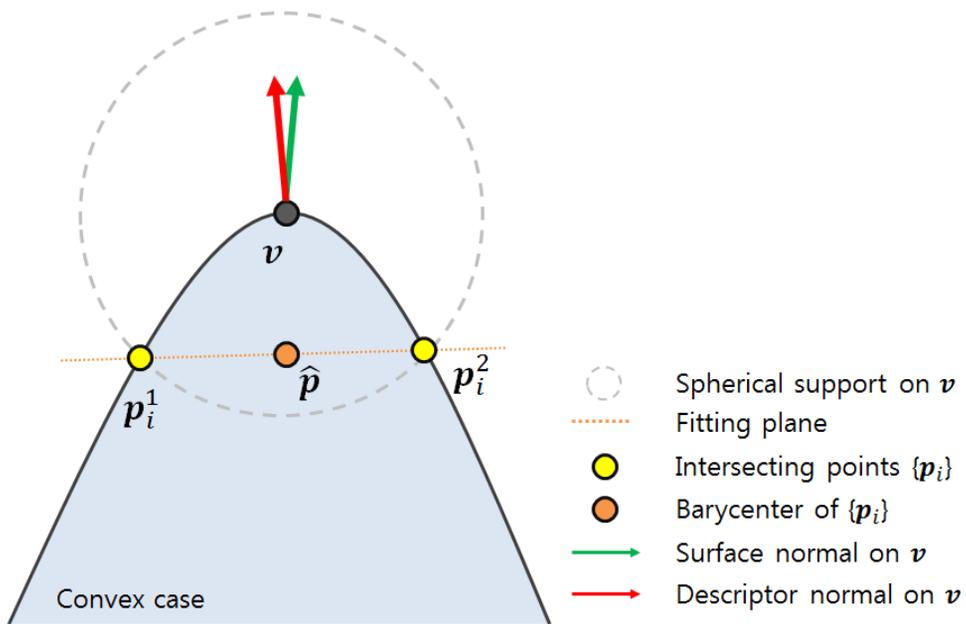


Fig. 3.9 Descriptor normal of convex vertex. Descriptor normal is along the surface normal.

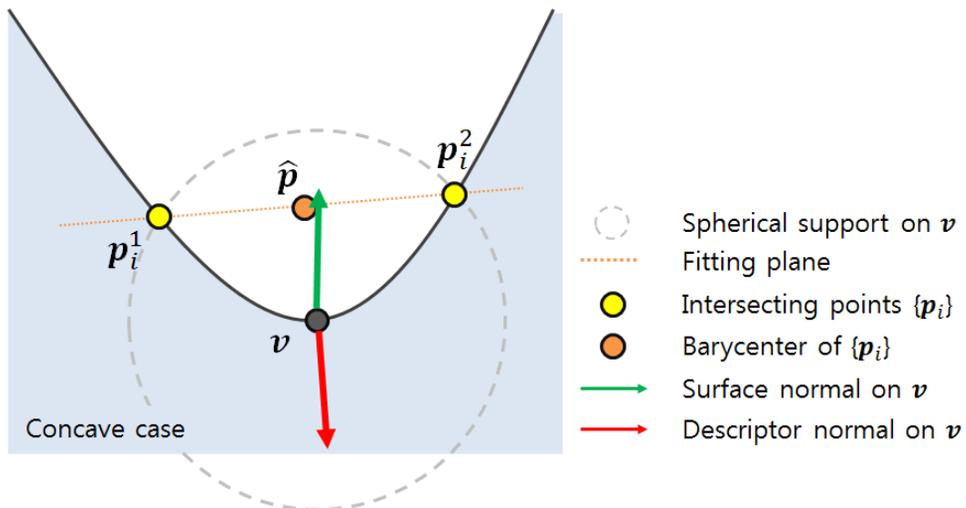


Fig. 3.10 Descriptor normal of concave vertex. Descriptor normal is opposite the surface normal.

### 3.3 Results and discussion

Fig. 3.11 shows the convex/concave analysis based on the normal variation within one-ring neighbors (top row) and the convex/concave analysis using the proposed method (bottom row). The convex region is represented by red, while the concave region is represented by blue. To show how well a surface signature on a fractured surface represents the characteristics of the surface effectively, we chose fractured surfaces that were counterparts to each other. In a complicated fractured surface, it can be seen that the distribution of convex and concave vertices from the conventional method can be simply regarded as noise.

Fig. 3.12 shows the convex region of one surface and the concave region of another surface transparently assuming that the fractured surfaces in Fig. 3.11 are matched with each other. Even though it is a simple method of flipping a 2D captured image of 3D distribution on a fractured surface, it can be said that the surface signature explains the shape characteristics of the fractured surface well through the large overlapping region between the fractured surfaces of the counterparts.

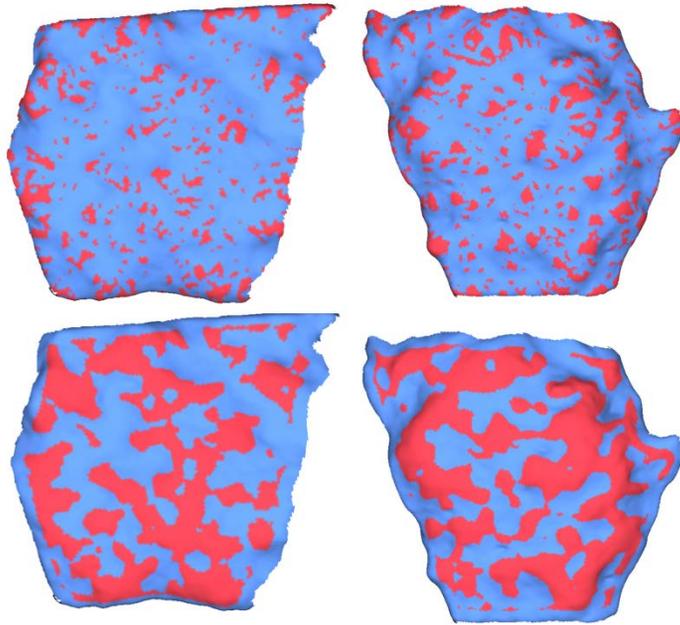


Fig. 3.11 Comparison of convex/concave analysis using one-ring neighbor (top) and using our method (bottom).

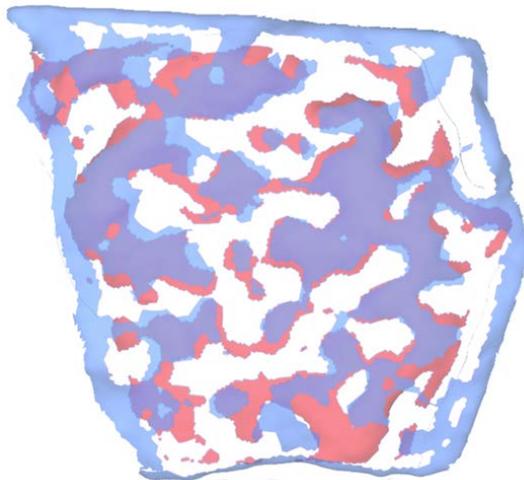


Fig. 3.12 Assuming the two surfaces of Fig. 3.11 are matched, the convex regions of fractured surfaces and the concave regions of other fractured surfaces are mostly overlapped (bottom).

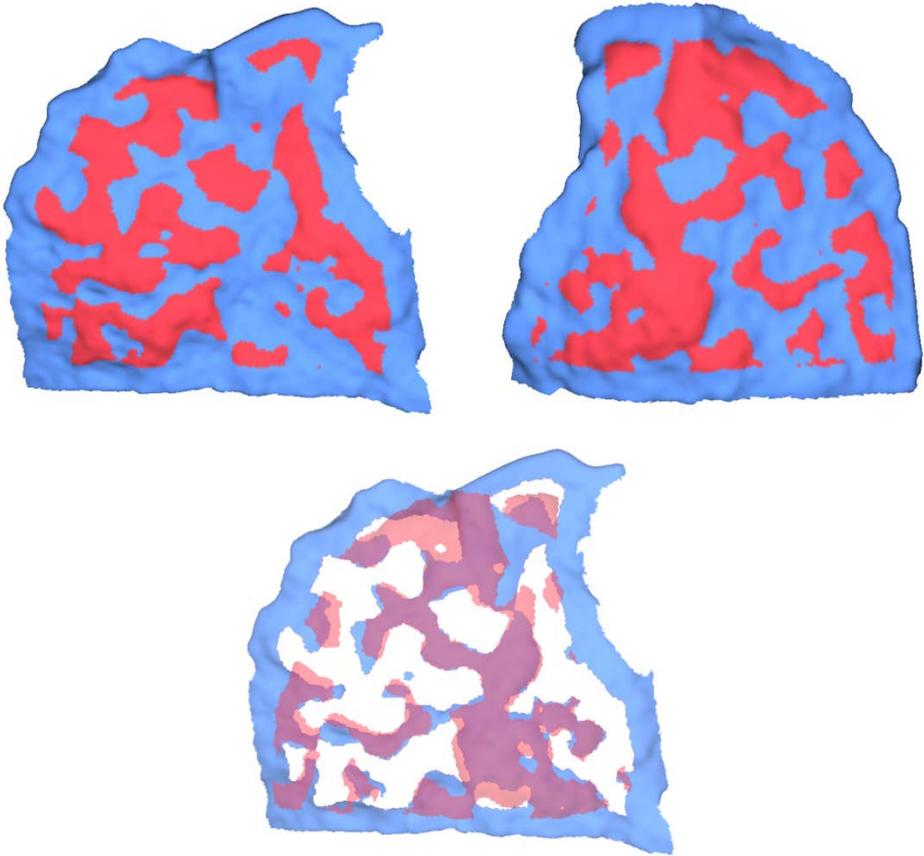


Fig. 3.13 Second surface signature example (top). Two surface signatures also have common regions on most surfaces (bottom).

Fig. 3.13 shows the surface signature and overlapping results for the fractured surfaces that form another counterpart. As in the previous example, we can see that most of the convex and concave regions from each fractured surface are overlapped.

If the previous examples are a 1: 1 matching case where two fractured surfaces are almost completely embedded, Fig. 3.14 is an example of a 1: N or partial matching case where two small fractured surfaces are matched to one larger fractured surface. In addition, Fig. 3.14 shows the overlapping results of the fractured surfaces in Fig. 3.13. Even though one fractured surface is a part of another fractured surface, the distribution of surface signatures is well overlapped.

Several practical examples demonstrate that our proposed surface signature is a good representation of the shape features of complex and noisy fractured surfaces and is suitable for use in matching.

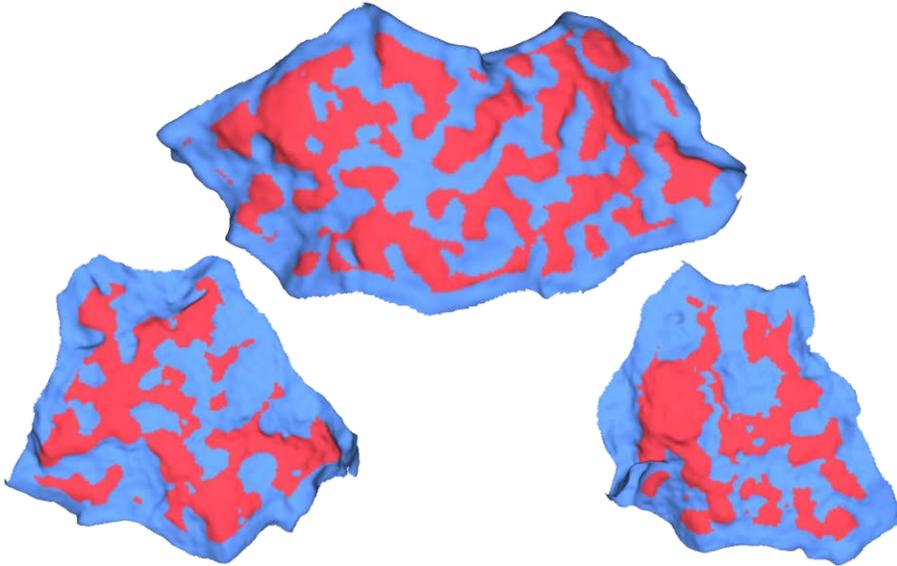


Fig. 3.14 Examples of case in which multiple small fractured surfaces match one large surface. The convex regions of each fractured surface are red, and the concave regions are blue.

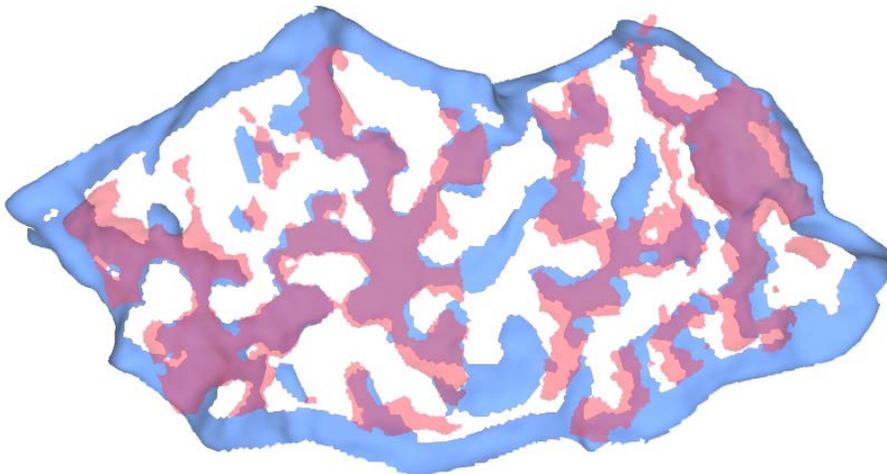


Fig. 3.15 Overlapped surface signatures in 1:N matching case. The small fractured surfaces' surface signature distributions are well overlapped with those of the large fractured surfaces.

# CHAPTER 4.

## REASSEMBLING

### 4.1 Overview

The proposed reassembling process of the broken object is shown in Fig. 4.1. The smoothed model is created by smoothing the given original object (Fig. 4.1 (a)) (Fig. 4.1 (b)). This smoothing model removes complex and noisy shapes from fractured surfaces to facilitate the subsequent segmentation process. The smoothed model is also applied to the calculation of roughness for fractured surface extraction. The smoothed model is segmented into several surface segments through the segmentation process (Fig. 4.1 (c)). Each surface segment is clustered into a fractured surface and an intact surface group by the roughness of the segment based on the curvature difference between the smoothed model and the original model (Fig. 4.1 (d)). Segments classified as fractured surfaces are used in the subsequent reassembling process. For this, a surface signature is calculated for each fractured surface (Fig. 4.1 (e)). In order to apply the distribution of the surface signature to the matching, a feature curve is created using the boundaries of the convex and concave regions. Based on the spin image and the curve information (position and direction of discrete curve point) defined on the feature curve, correspondence and transformation between two fractured surfaces are obtained (Fig. 4.1 (f)). In the matching candidate between the obtained fractured surfaces,

the similarity is found to be the best, and the final reassembled result can be obtained by applying the transformation (Fig. 4.1 (g)).

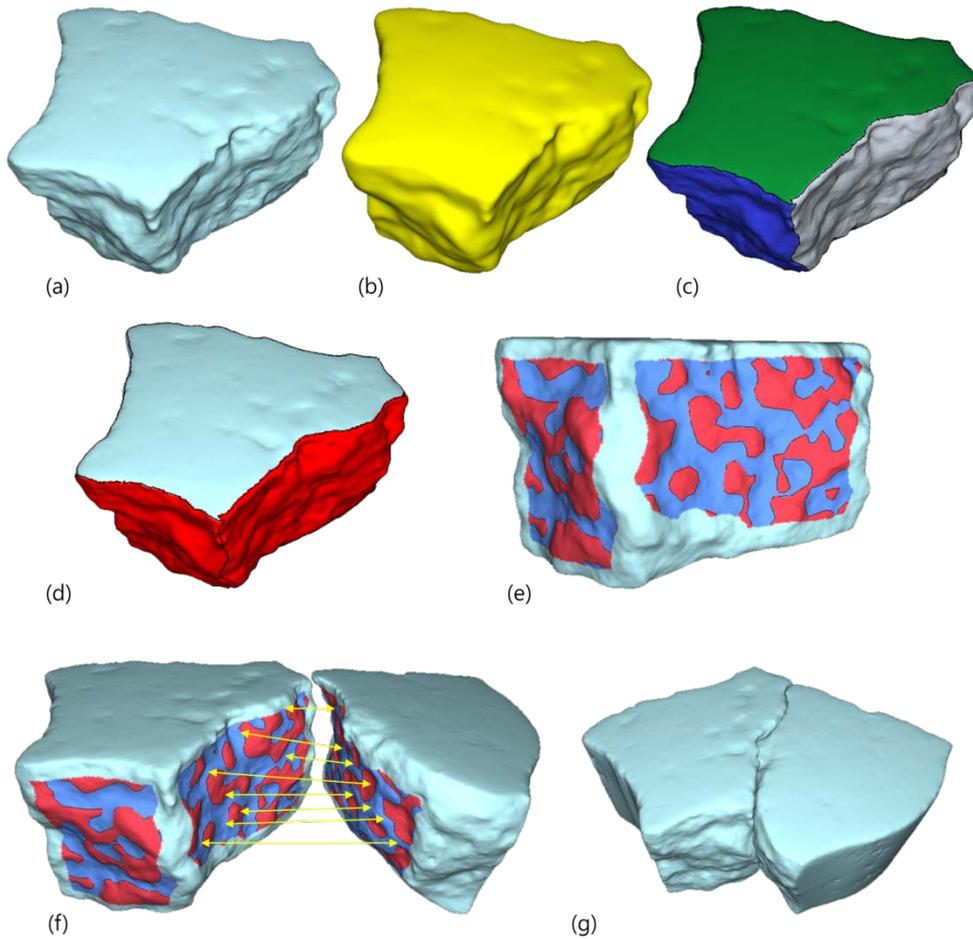


Fig. 4.1 The overview of our reassembling process. (a) Original fragment, (b) smoothing model, (c) segmentation, (d) fractured surface clustering, (e) surface signature calculation, (f) correspondence detection between fractured surfaces, and (g) matching.

## **4.2 Fractured surface extraction**

In order to reassemble a broken object, the fractured surface on which the matching is made must first be defined. Generally, a fractured surface is rough compared to a surface that has not been broken, and forms an angle of some degree with the surfaces forming the boundary due to the external force or pressure that causes the break. Based on these features, the fractured surfaces of broken objects are extracted through segmentation and surface roughness analysis.

### **4.2.1 Smoothing**

The most distinctive feature of a broken object is that the noisy surface is the most important part of the fractured surface. In fact, mesh-smoothing or mesh-denoising techniques have been studied with the aim of smoothing uneven surfaces. However, in the reassembly of a fractured surface, we try to emphasize the features of the fractured surface and make the extraction of the fractured surface more effective through mesh smoothing.

The initial Laplacian smoothing [35] simply involved moving one vertex on the mesh to the average position of its neighboring vertices. It is a very easy and simple algorithm, but the problem is that the shape shrinks as smoothing is repeated. To solve the shrinking problem, two-step smoothing based on signal processing and fast Laplacian smoothing has been proposed [36]. Although a variety of smoothing technologies have been proposed, ranging from anisotropic

smoothing to preserve features [37-40] and bilateral filtering applied to 3D meshes [41] to smoothing using normal rather than vertex positions [42, 43], the aim of smoothing is still preventing shape shrinkage and preserving features.

Smoothing in the reassembly of a broken object aims to further emphasize the fractured surfaces that can be ignored in previous smoothing methods. In detail, the smoothed model of the original broken object preserves the feature that is the boundary between the fractured surface and the intact surface and relaxes the shape of the fractured surface. This enables us to make the segmentation process more effective and accurate. In addition, based on the smoothed model, we can evaluate the roughness by comparing it with the original model. This roughness is utilized as the basis for fractured surface extraction.

To achieve this goal, we use adaptive smoothing proposed by Lavoué [44]. Lavoué's adaptive smoothing is easier to implement than other smoothing methods, does not suffer from shrinking, and has the advantage of adjusting the degree of noise to smoothing by a support radius. The fact that the goal of segmentation after smoothing is only the extraction of the fractured surface and that the exact segmentation of the other intact surface is not so important is one of the reasons for choosing adaptive smoothing.

Adaptive smoothing is based on Taubin's two-step smoothing [36]. A smoothed position  $\mathbf{v}'$  of a vertex  $\mathbf{v}$  can be determined by:

$$\mathbf{v}^m = (\mathbf{v} + \lambda \cdot s(\mathbf{v})) \quad \text{with } s(\mathbf{v}) = \frac{1}{n} \sum_{\mathbf{e}_i \in x} (\mathbf{e}_i - \mathbf{v})$$

$$\mathbf{v}' = (\mathbf{v}^m + \mu \cdot s(\mathbf{v}^m))$$

where  $\mathbf{e}_i$  is the  $i^{th}$  edge point of the local window of vertex  $\mathbf{v}$ , and the constants  $\lambda$  and  $\mu$  are 0.6307 and -0.6732, respectively.

The following Fig. 4.2 is an example of smoothing several broken objects, and it can be seen that the fractured surface area is remarkably deformed.

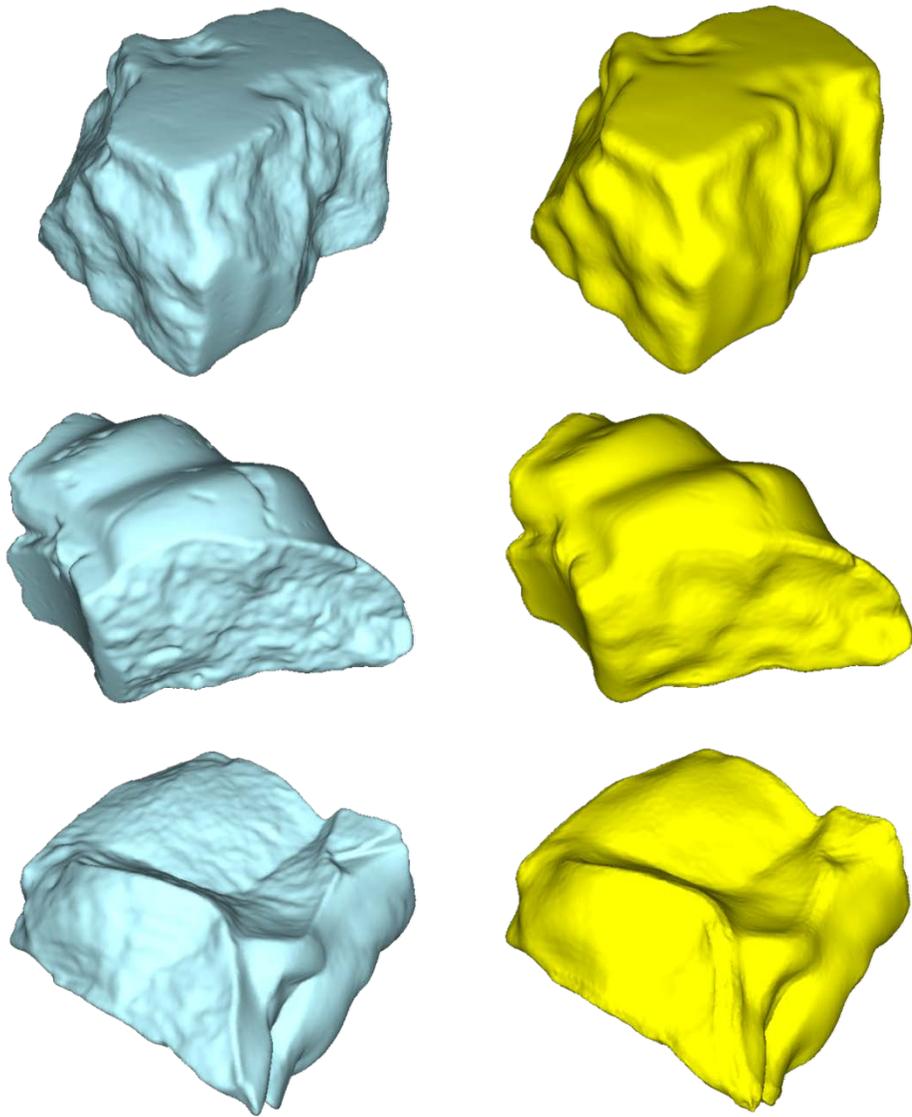


Fig. 4.2 Examples of mesh smoothing. The left column is the original broken object mesh, and right the column is the smoothed model.

## 4.2.2 Roughness

The most noticeable difference between a fractured surface and an intact surface in a broken object is the roughness of the surface. As we can select a curved part to match a 2D jigsaw puzzle rather than a planar part, we select an uneven surface rather than a smooth surface instinctively when assembling a broken 3D object. A measure for representing the roughness of a vertex in a 3D mesh was introduced by several researchers. Karni and Gotsman [45] introduced the Geometric Laplacian (GL) to measure the smoothness of the vertex  $\mathbf{v}$ :

$$GL(\mathbf{v}) = \mathbf{v} - \frac{\sum_{i \in n(\mathbf{v})} l_i^{-1} \mathbf{v}_i}{\sum_{i \in n(\mathbf{v})} l_i^{-1}}$$

where  $n(\mathbf{v})$  is a set of indices of the neighbors of  $\mathbf{v}$ , and  $l_i$  is the Euclidean distance from  $\mathbf{v}$  to  $\mathbf{v}_i$ . GL measured the roughness of the vertex using a difference vector between the vertex  $\mathbf{v}$  and the new position of the  $\mathbf{v}$  after Laplacian smoothing. With the same idea, Gelasca et al. proposed a global roughness variation [46]. They defined the global roughness using variations of the geometric distance between the 3D model and its smoothed model. Another researcher proposed a roughness measure using dihedral angles of edges sharing vertices [47, 48]. Lavoué [44] proposed a roughness measure using the curvature difference between the 3D model and the smoothed model (see Fig. 4.3). He used a feature-preserving smoothing method so that smoothing rarely affects the roughness of salient and sharp features (i.e., edges, ridges). He also adopted the local windows to make the roughness measure independent from mesh connectivity

and control the noise scale factor.

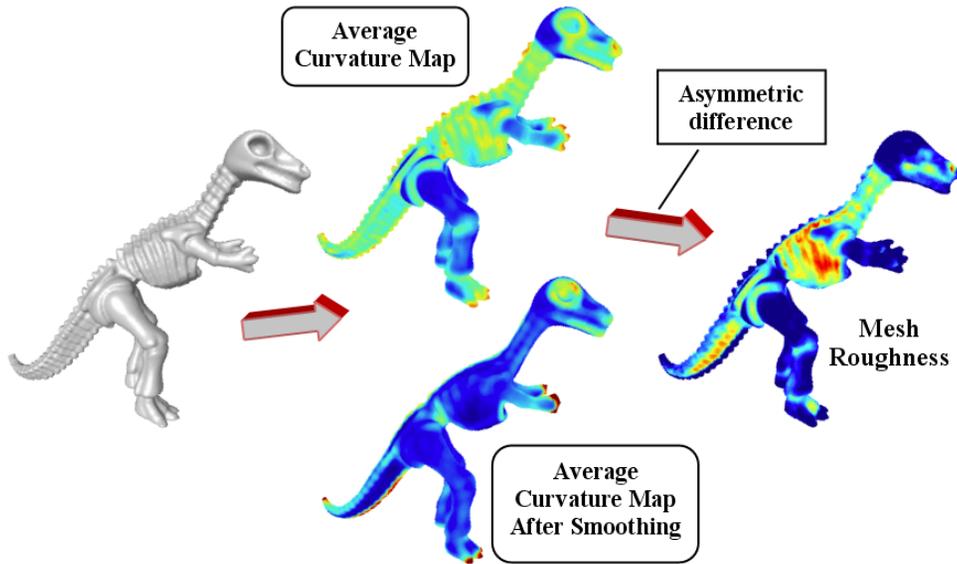


Fig. 4.3 Overview of Lavoué's roughness calculation algorithm [44].

It is obvious that the smoothing effect of smoothed fragments of a broken object is larger in a relatively noisy fractured surface area, so we propose a roughness measure suitable for a broken object by exploiting Lavoué's method. The roughness of the vertex is essentially determined by the curvature difference between the original model and the smoothed model. At this time, we make the local window vary according to the scale factor that defines the noise of the mesh (Fig. 4.4). The local window serves as a sphere of radius  $r$  centered on the vertex  $v$

and plays a role in the robustness to the connectivity and density of the mesh. The  $r$  value at this time is a scale criterion for defining noise. Between the defined local window and the mesh, we can get two sets of inner and edge points. The inner point is the vertex inside the sphere of the mesh (Fig. 4.4 blue dots), and the edge point is the intersection point between the sphere and the mesh of the local window (Fig. 4.4 green dots). The curvature of a vertex  $\mathbf{v}$  on the mesh is the average of the curvature of the inner point  $\mathbf{v}_i$  and the curvature of the edge point  $\mathbf{v}_e$ :

$$C(\mathbf{v}) = \frac{1}{n} \sum_{j=1}^n C(\mathbf{v}_i)_j + \frac{1}{m} \sum_{k=1}^m C(\mathbf{v}_e)_k$$

where the curvature of the edge point is the interpolation of the curvature on each end vertex:

$$C(\mathbf{v}_e) = \frac{d_2}{d_1 + d_2} C(\mathbf{v}_1) + \frac{d_1}{d_1 + d_2} C(\mathbf{v}_2).$$

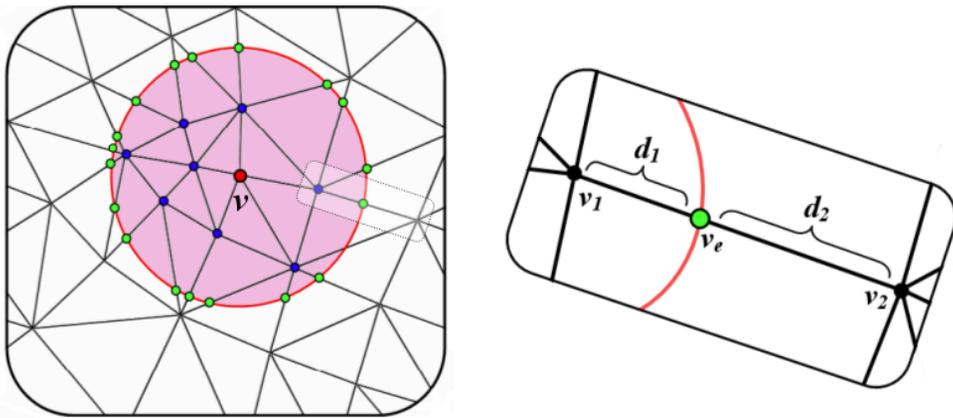


Fig. 4.4 Local windows on vertex  $\mathbf{v}$  (left) and curvature interpolation on edge points (right) [44].

The roughness of a vertex  $\mathbf{v}$  based on the above curvature is as follows:

$$R(\mathbf{v}) = Cr^o(\mathbf{v}) - Cr^s(\mathbf{v})$$

where  $Cr^o(\mathbf{v})$ ,  $Cr^s(\mathbf{v})$  is the vertex curvature of  $\mathbf{v}$  of the original model and the smoothed model, respectively.

However, sharp regions, such as edges or ridges, cannot avoid deformation in the smoothing process despite the feature-preserving method (see Fig.4.5 (a)). This deformation leads to an increase in the curvature difference in sharp regions and causes an ambiguous situation in fractured surface selection.

To mitigate this problem, we define curvature-weighted vertex roughness as follows:

$$R(\mathbf{v}) = \gamma_{\mathbf{v}} * (Cr^o(\mathbf{v}) - Cr^s(\mathbf{v}))$$

$$\gamma_{\mathbf{v}} = 1 - \frac{(Cr_{m}^o(\mathbf{v}) - Cr_{min}^o)}{(Cr_{max}^o - Cr_{min}^o)}$$

where  $Cr_{max}^o$ ,  $Cr_{min}^o$  are the maximum and minimum curvatures on the original model, respectively. These values are taken through a 90% winsorization process, in which all curvatures below the 5<sup>th</sup> percentile are set to the curvature of the 5<sup>th</sup> percentile, and all curvatures above the 95<sup>th</sup> percentile are set to the curvature of the 95<sup>th</sup> percentile. This winsorization can reduce the effect of spurious outliers by limiting extreme values. The weight plays a role by reducing the change of curvature for a high-curvature vertex and increasing the contrast of roughness between surface segments.

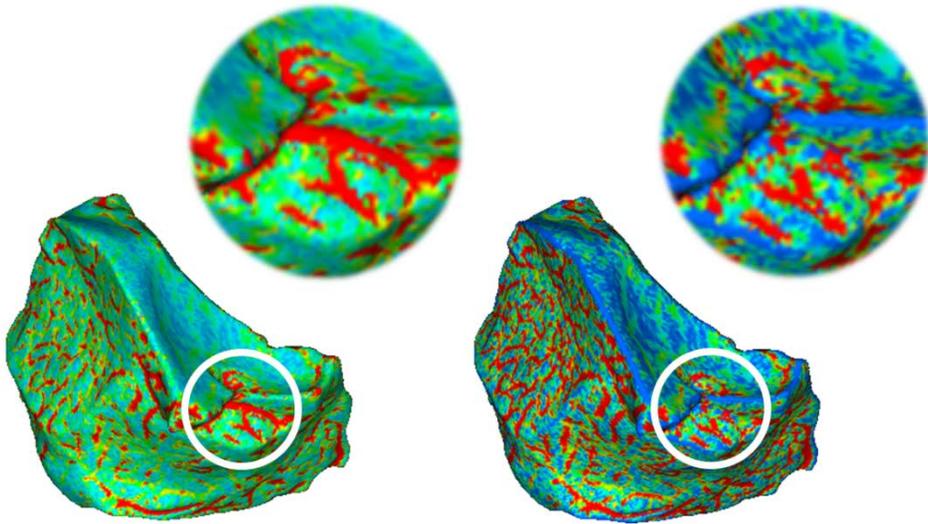


Fig. 4.5 Vertex roughness as curvature difference between original and smoothed model (a). Curvature-weighted roughness (b).

As shown in Fig. 4.5, the curvature-weighted roughness makes the roughness of the ridge and edge of the object smaller and the roughness of the area of relatively small curvature larger, thereby further ensuring the roughness contrast between the fractured surface and the intact surface.

### 4.2.3 Segmentation

Curvature is the simplest geometric feature for mesh segmentation. In addition, as shown in Fig. 4.5, high curvature regions can represent boundaries separating the object into intact surfaces and fractured surfaces. Thus, we apply watershed-based mesh segmentation using the surface curvature value for the height function [27].

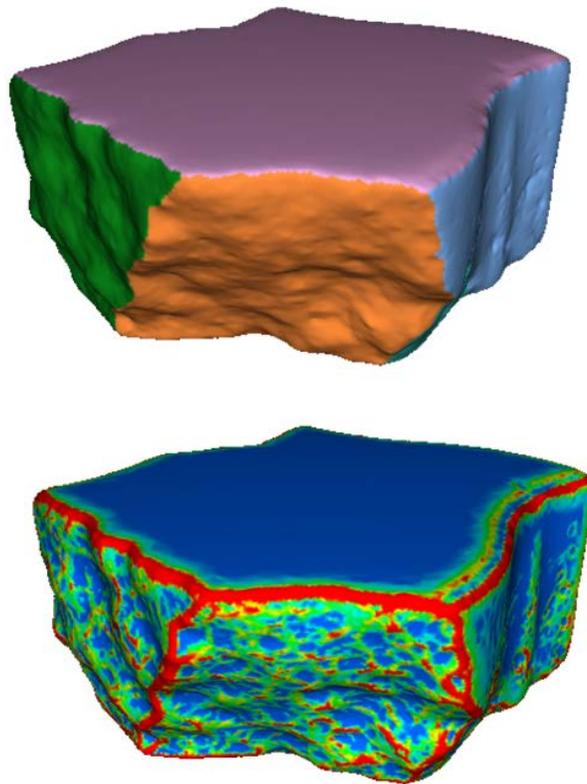


Fig. 4.6 Mesh segmentation and mean curvatures

First, curvature values at all vertices are calculated [49]. Then, all local minimum vertices are found, and each vertex is uniquely labeled. Next, from the unlabeled vertices, all passing vertices following the path of steepest gradient descent are stored in a stack until a labeled region is encountered. After every vertex has been labeled, the last step of segmentation is merging the over-segmented regions. In this paper, we additionally consider curvature value. We defined  $C_{avg}$ , the average boundary curvature of segment  $S$ , as follows:

$$C_{avg}(S) = \frac{\sum_{v \in B(S)} C(v)}{\sum_{v \in B(S)} 1}$$

Assume that there are two segments  $S_a, S_b$ . If  $C_{avg}(S_a + S_b) < (C_{avg}(S_a) + C_{avg}(S_b)) * 0.5$ , it means that the two segments are on opposite sides of the high curvature region. Therefore, those regions are not merged. In this way, a merged segment whose boundary is in the high curvature region is obtained.

#### 4.2.4 Fractured surface clustering

The roughness of the surface segment  $S$  is calculated as the average of the roughness  $R(\mathbf{v}_i)$  of the vertices  $\mathbf{v}_i$  included in the segment:

$$R_{seg}(S) = \frac{1}{N} \sum_{\mathbf{v}_i \in S} R(\mathbf{v}_i)$$

where  $N$  is the total number of vertices included in segment  $S$ .

After calculating all of the roughness values of the segments obtained by the segmentation, the fractured surface can be obtained easily by clustering. There are three types of clusters to find: intact, ambiguous, and fractured. Ambiguous clusters are added to account for the effects of unexpected segmentation errors or distortion on intact surfaces. Fractured surfaces and ambiguous surfaces are used in the subsequent matching process.

Since the number of clusters is already set to three, and clustering data has only 1D segment roughness, it can be calculated simply by K-means clustering [50]. The K-means clustering method determines the number of  $k$  random seeds in the data and performs initial clustering by including similar data in the same cluster based on the similarity between the remaining data and the seed. When the first clustering is complete, the center of the data in the cluster is recalculated, and clustering is performed again by designating these centers as a new seed. This process is repeated until the cluster is no longer changed to obtain the final clustering result (see Fig. 4.7).

The lower part of Fig. 4.8 shows the result of clustering. The intact surface is blue,

the ambiguous surface is green, and the fractured surface is red.

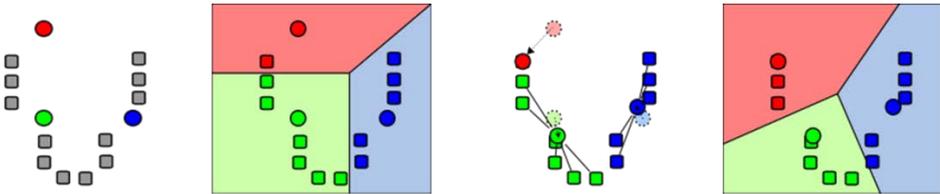


Fig. 4.7 K-means clustering algorithm with  $k=3$  [51].

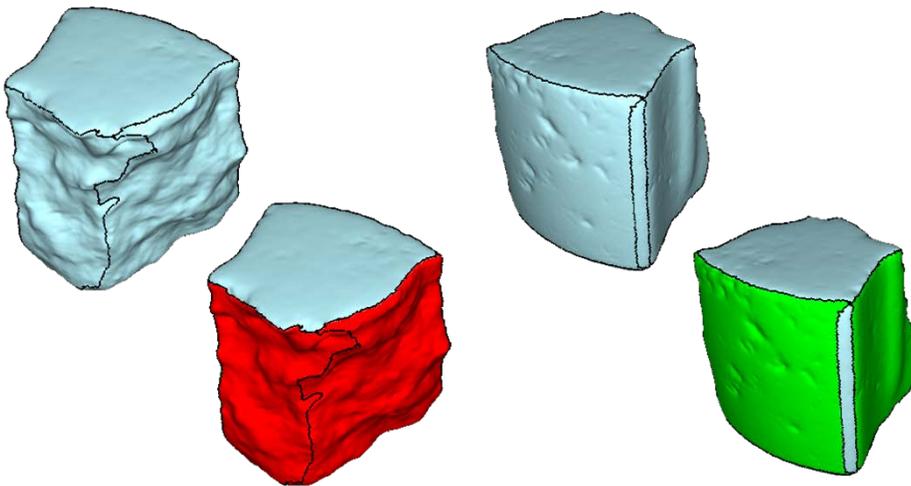


Fig. 4.8 Results of segment clustering(bottom row). Red surface segments indicate the fractured surface and green surface segments indicate ambiguous surface.

## 4.3 Fractured surface matching

### 4.3.1 Feature curve

In order to match two fractured surfaces, discrete feature curves are extracted using surface signature information.

As mentioned earlier, the surface signature of a vertex on a surface indicates whether the vertex is convex. That is, the surface signature can divide a fractured surface into convex and concave regions. Therefore, a discrete feature curve that describes the geometric characteristics of a fractured surface can be defined as the boundary between convex and concave regions.

Feature curves are composed of points that satisfy the following:

$$p_{\text{curv}} = \left\{ p_{\text{curv}}^i = \frac{(p_{e_i}^1 + p_{e_i}^2)}{2} \right\}$$

where  $p_{e_i}^1, p_{e_i}^2$  are endpoints of edge  $e_i$

$$\text{and } \text{Sig}(p_{e_i}^1) * \text{Sig}(p_{e_i}^2) = -1.$$

By locating a particular region (convex or concave) on one side of the curve, the direction of the curve is defined consistently for any fractured surfaces. In addition, this consistent directional curve is applied to estimate the similarity of two aligned fractured surfaces.

Extracted discrete feature curves include every boundary vertex dividing a fractured surface based on surface signatures, so they include unnecessary vertices.

Since this redundancy makes the matching process ineffective, curves need to be simplified properly. In this paper, we simplify feature curves and leave some feature points that can adequately represent curves using the Douglas-Peucker simplification method [52].

The main algorithm of this method involves finding the key point. At the beginning of simplification, a line segment is generated by connecting two end points of a single discrete curve. The key point is the farthest curve point from the line segment. After finding the key point, the curve is split into two separate segments, and a new key point is detected for each segment by repeating the previous process. The simplification process continues until a newly detected key point is closer to the current line segment than the predefined threshold. This simplification method is well illustrated in Fig. 4.9.

Curve points extracted by the simplification method are considered feature points that provide important shape information of the fractured surface to find correspondences between two fractured surfaces in the following matching process.

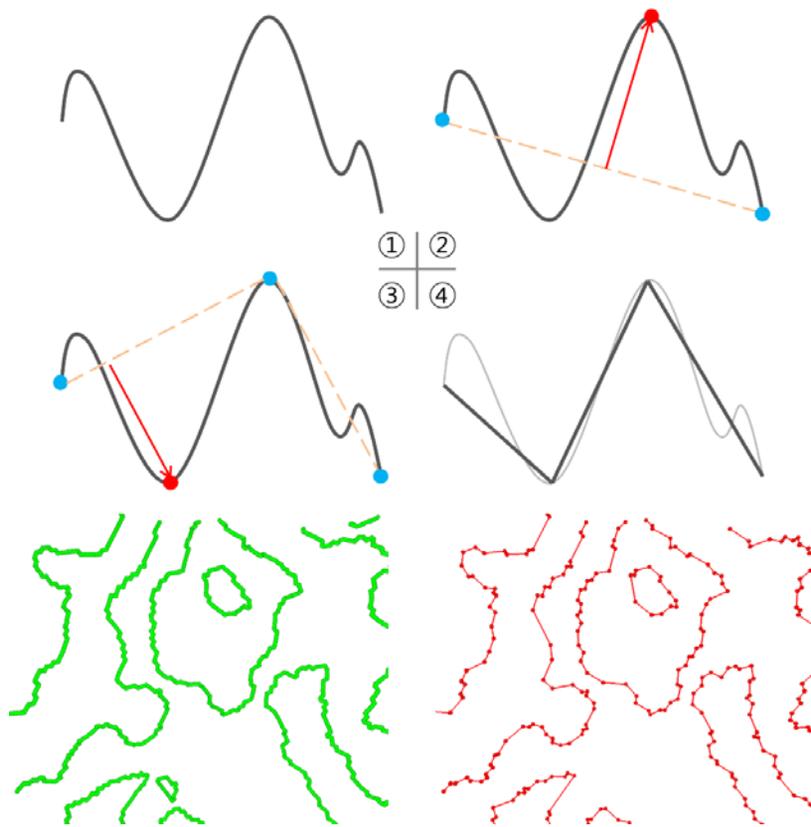


Fig. 4.9 Douglas-Peucker curve simplification and its results.

### 4.3.2 Matching

In the reassembling application for broken objects, one fragment can have an arbitrary number of fractured surfaces. Thus, the matching between fragments can be regarded as the matching between fractured surface sets of fragments, and this matching can lead to the most well aligned fractured surface pair among every possible fractured surface pair. For this reason, the matching process of the reassembling application is based on the pairwise matching of the fractured surface. Pairwise matching between two fractured surfaces essentially uses spin images of feature points extracted in the previous section. A spin image descriptor, proposed by Johnson, is a 2D representation of a 3D object [53, 54]. Assume there is an oriented point  $O$  on the 3D surface, which has position  $p$  and surface normal  $n$ , as shown in Fig. 4.10. Then, an oriented point basis is defined using the tangent plane  $P$  through  $p$  oriented perpendicularly to  $n$  and the line  $L$  through  $p$  parallel to  $n$ . A position of point  $x$  around  $O$  can be represented as two coordinates of the basis  $\alpha$ , the perpendicular distance to the line  $L$ , and  $\beta$ , the signed perpendicular distance to the plane  $P$ . In other words, a spin image with an oriented point  $O$  is the function that projects 3D points  $x$  to the 2D coordinates, which is invariant to transformation.

$$S_O: R^3 \rightarrow R^2$$

$$S_O(x) \rightarrow (\alpha, \beta)$$

$$(\alpha, \beta) = (\sqrt{\|x - p\|^2 - (n \cdot (x - p))^2}, n \cdot (x - p))$$

The similarity of the two spin images P and Q is estimated as follows:

$$C(P, Q) = (\operatorname{atanh}(R(P, Q)))^2 - \lambda \left( \frac{1}{N - 3} \right).$$

Here,  $R(P, Q)$  is the normalized linear correlation coefficient,  $\lambda$  is an expected number of overlapping pixels, and  $N$  is a number of actually overlapping pixels during the calculation of  $R(P, Q)$ . The similarity measure returns a high value for two highly correlated images.

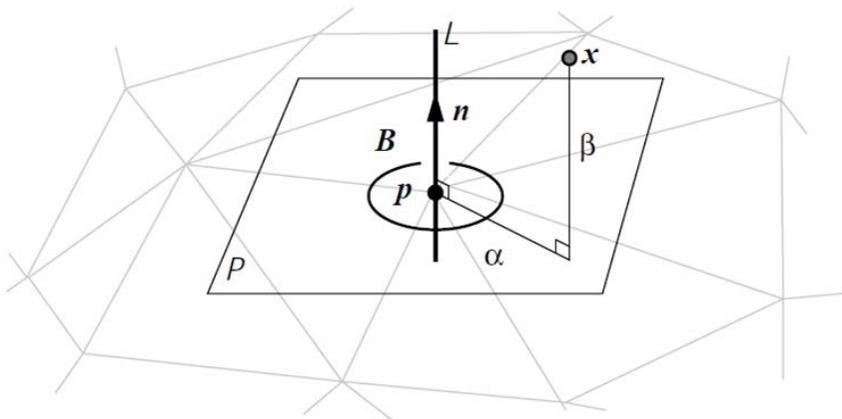


Fig. 4.10 Oriented point basis for spin image [53].

By comparing the spin images of fractured surfaces, we can construct a set of possibly matching correspondences between two fractured surfaces. The RANdom Sample Consensus(RANSAC) method is applied to these candidate data to estimate a transformation that aligns the two fractured surfaces properly. RANSAC is an iterative method that can derive the most consistent answer with a dataset including outliers [55].

The RANSAC method is applied as follows for fractured surface matching. Suppose there are two fractured surfaces: fixed surface  $S_F$ , which is a matching reference, and moving surface  $S_M$ , which will be transformed onto the fixed surface. Through the spin image matching process, two points from both  $S_F$  and  $S_M$ , which have similar spin images, are paired as a correspondence.

First, three correspondences are selected randomly among the possibly matching correspondence dataset. In addition, the transformation matrix, which aligns each corresponding point simultaneously, is calculated. If the selected correspondences are consistent with the other remaining correspondences, other points from the moving surface should be located close to the corresponding points from the fixed surface after applying the current transformation.

If there is a corresponding point pair whose distance between two points after transformation is applied is smaller than the given threshold, we can say that the correspondence is in consensus with the current randomly selected correspondences. When a larger number of correspondences is selected with the currently selected correspondences, a more reliable alignment can be conducted by

the current transformation. RANSAC finds the correspondence set and its transformation iteratively, maximizing the number of consensuses.

As mentioned at the beginning of this section, matching between two fragments involves finding an optimal matching pair among all possible pairs between two fractured surface sets from each fragment. However, the result from RANSAC with a spin image, which is determined by the maximum consensus number, is insufficient to find the best matching fractured surface pair. One of the reasons is that the complicated and noisy fractured surface weakens the spin image descriptor. This is because the local descriptor is sensitive to noise, so it does not guarantee robust matching. The other reason is the fact that a number of consensuses can be affected by the size of the correspondence candidates. A more reliable matching pair could have a small number of consensuses because the fractured surface is small.

Therefore, in this paper, we defined a new similarity measure to find the correct matching pair that includes discrete feature curve information from the surface signature. The similarity measure  $f(S_F, S_M)$  between the fractured surfaces  $S_F, S_M$  is defined by

$$f(S_M, S_s) = \frac{f_{dist} * f_{nor}}{N_{con}}$$

where  $N_{con}$  is the number of consensuses from RANSAC. Here, similarity measures  $f_{dist}$  and  $f_{nor}$  are added.  $f_{dist}$  indicates the distance between two feature curves of two surfaces, and  $f_{nor}$  indicates the normal deviation between

close feature curve points.

As shown in Fig. 11, to calculate the distance between two curves, the point-to-curve distance is more reasonable than the point-to-point distance. Thus, in this paper, to calculate the distance between two curves, all feature points  $\{p_1, \dots, p_n\}$  from the feature curves of fractured surface  $S_F$  are projected to the feature curves of  $S_M$ . Then, among all projected point pairs, only point pairs whose distance between two points is smaller than the given threshold are selected. As a result, we obtain close feature point pairs:  $\{(p'_1, q'_1), \dots, (p'_m, q'_m)\}$ .

Using these point pairs, the distance measure  $f_{dist}$  between fractured surfaces  $S_F$  and  $S_M$  is defined as follows:

$$f_{\text{dist}}(S_F, S_M) = \frac{1}{m} \sum_{i=1}^m \|p'_i - q'_i\|$$

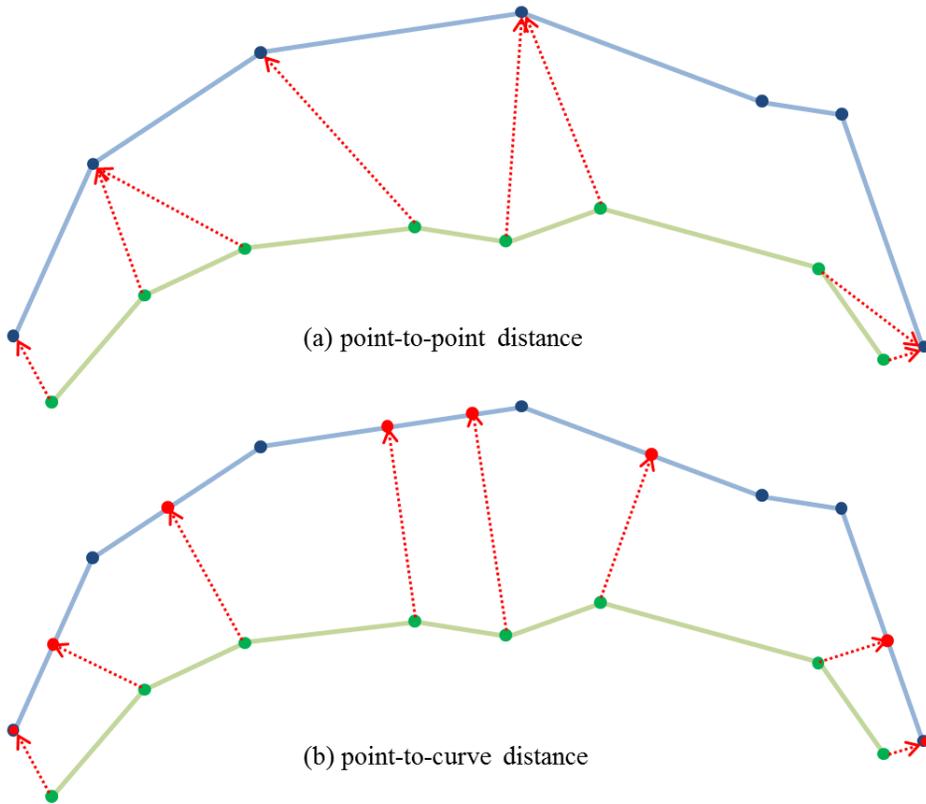


Fig. 4.11 Curve distance metric. Point-to-point distance (a) and point-to-curve distance (b)

In order to calculate another similarity measure  $f_{\text{nor}}$  we calculate curve normal first. For a discrete feature curve, surface normal and curve progress direction are known. Especially in generation of feature curve, curve direction is generated consistently for any fractured surface. So the curve normal can be calculated simply by cross product surface normal and curve direction. Considering a matching of fractured surface, two fractured surfaces are aligned facing each other and the convex region of one fractured surface is fitted to the concave region of the

other fractured surface. Because the direction of curve is consistent, curve normals should have same direction in correct alignment case. Based on these characteristic, curve normal measure  $f_{\text{nor}}$  is defined as

$$f_{\text{nor}}(S_M, S_S) = \frac{1}{m} \sum_{i=1}^m 1 - (\mathbf{n}_{\text{curv}}^{p'_i} \cdot \mathbf{n}_{\text{curv}}^{q'_i})$$

Newly defined similarity measure for matching of two fractured surfaces has low values when two surfaces are counterpart surface each other and well aligned.

## **CHAPTER 5.**

# **EXPERIMENTS**

We applied the proposed descriptor and matching algorithm to real-world broken objects shown in Fig. 5.1. The 3D digital models of broken objects were created by the 3D laser scanning method used in [14]. The broken brick model in Fig. 5.1 consists of six fragments. The fractured surfaces between brick fragments have an almost planar shape. The second example is the broken cake model in Fig. 5.2. The broken cake models are composed of 11 fragments. Curved surfaces surround the cake models, and some fragments have combinational matching relationships that show partial matching. The fractured sculpture model is shown in Fig. 5.3, and it has 15 fractured parts. The complicated shapes of the sculpture and uneven fragments make it the most challenging problem.

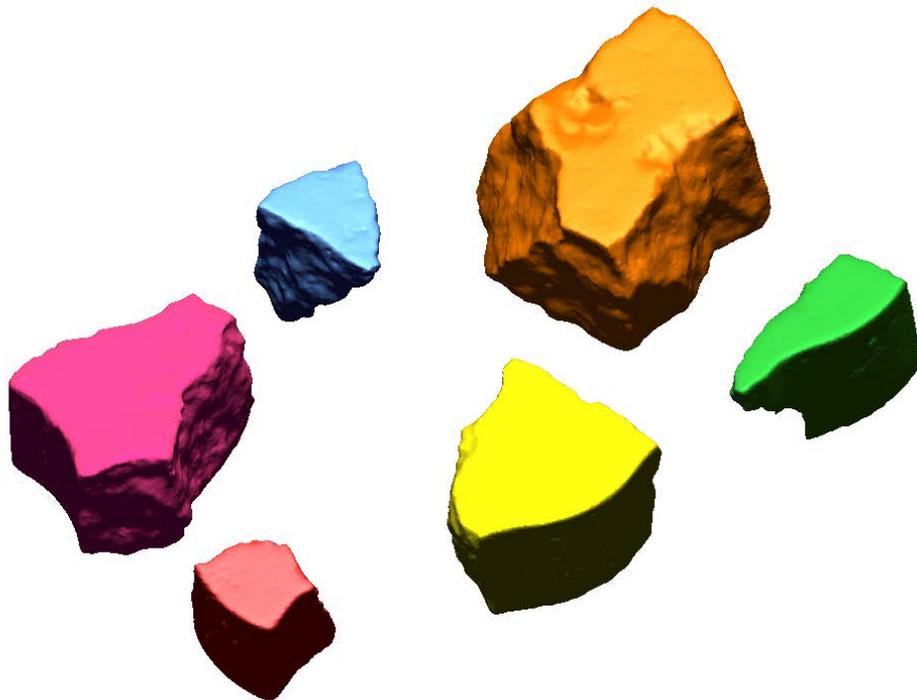


Fig. 5.1 Broken brick models.



Fig. 5.2 Broken cake models.

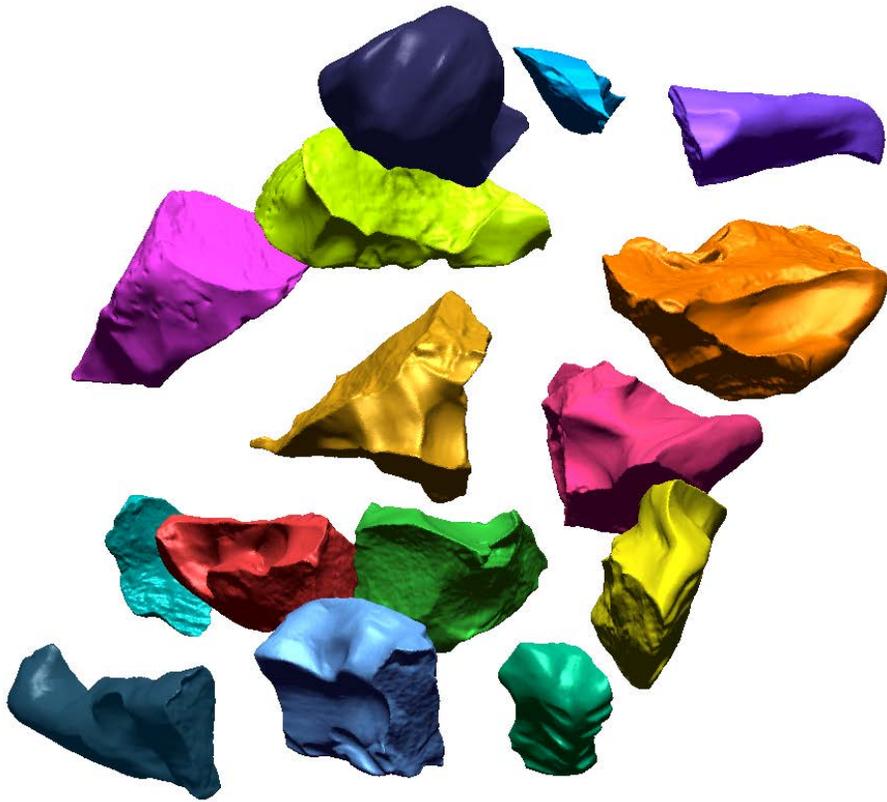


Fig. 5.3 Broken sculpture models.

Pairwise matching of two fragments was evaluated first. Two fragments were selected from the fragment set, and feature curves and spin images were extracted for each fragment. Only fractured surfaces that are clustered by roughness were considered in matching. All experiments were conducted on a 3.40-GHz machine. Details of fragments and numerical results of pairwise matching are shown in Table 5.1.

Table 5.1 Numerical results of pairwise matching.

Test No.	Number of faces		Number of segments		Number of feature points		Timing (in sec.)	
	A	B	A	B	A	B	Feature Gen.	Matching
1	265,792	176,166	10	5	2,563	1,261	10.928	15.522
2	287,450	281,586	8	9	2,295	1,884	15.023	20.229
3	182,152	136,142	10	13	1,782	1,814	9.913	15.138
4	181,330	160,094	12	7	1,983	1,646	11.859	15.341
5	122,292	125,518	14	19	2,535	618	6.377	7.283
6	182,958	151,058	15	12	1,491	1,672	11.41	12.939
7	136,554	77,828	14	7	894	472	6.769	2.003

Figs. 5.4–5.6 illustrate the results of the pairwise matching experiments. Fig. 5.4 shows the results of pairwise matching of brick fragments. In the first experiments, Fragments A and B have three and two fractured surfaces, respectively. The transformation matrix is estimated from the match with the minimum similarity measure among all possible matching pairs. Fig. 5.4-(d) shows that the proposed descriptor and matching algorithm can achieve reasonable matching despite the data loss between the fractured surfaces.

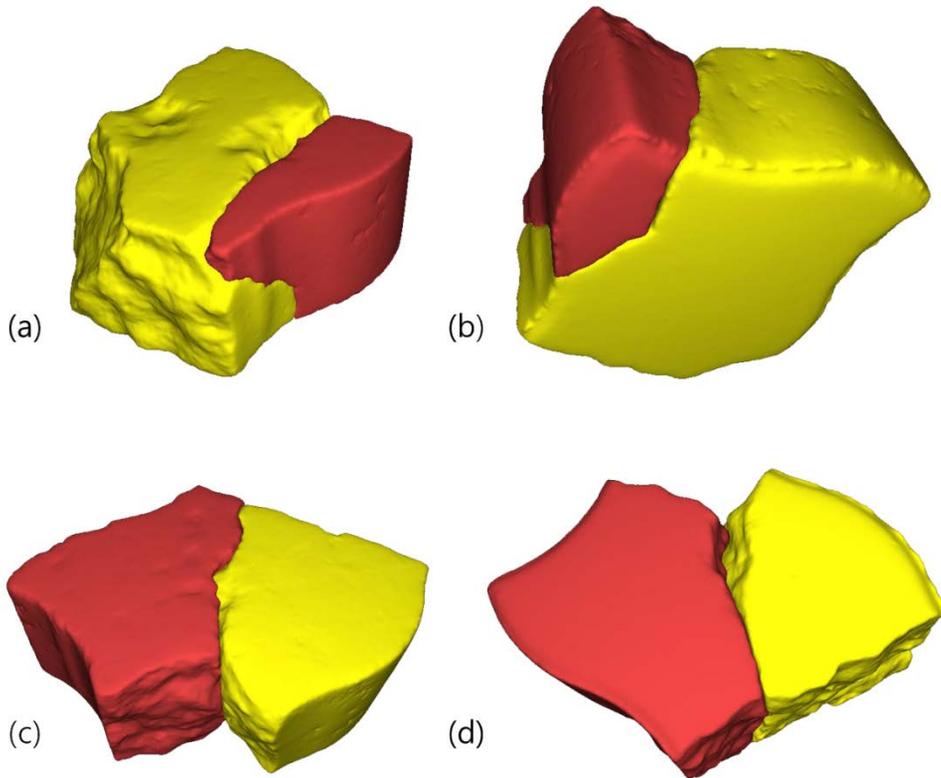


Fig. 5.4 Pairwise matching for brick fragments. (b), (d) are the difference views of (a), (c), respectively.

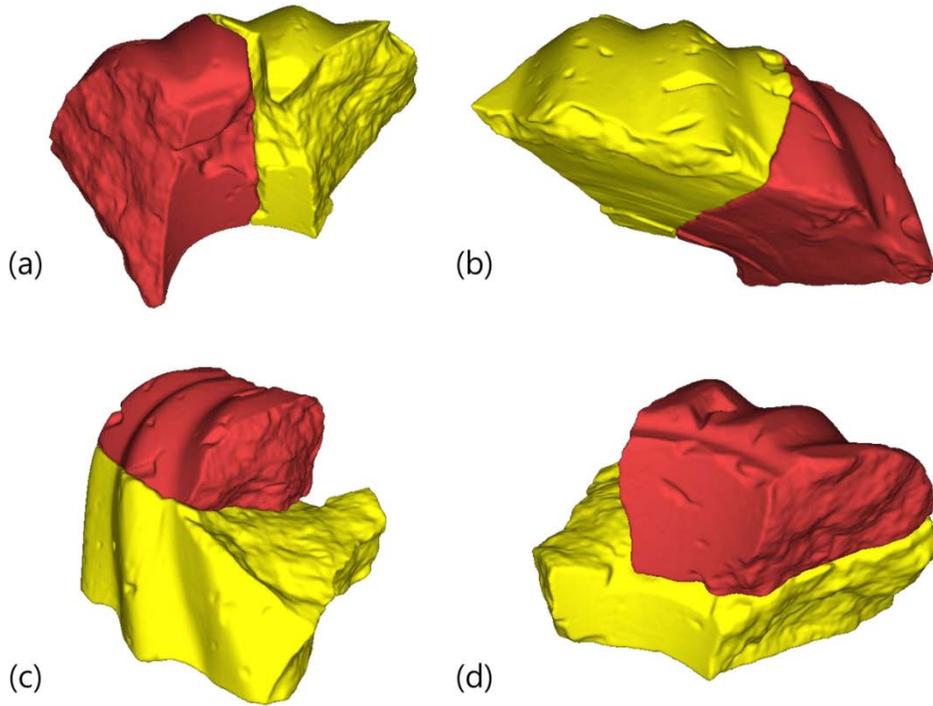


Fig. 5.5 Pairwise matching for cake fragments. (b), (d) are the difference views of (a), (c), respectively.

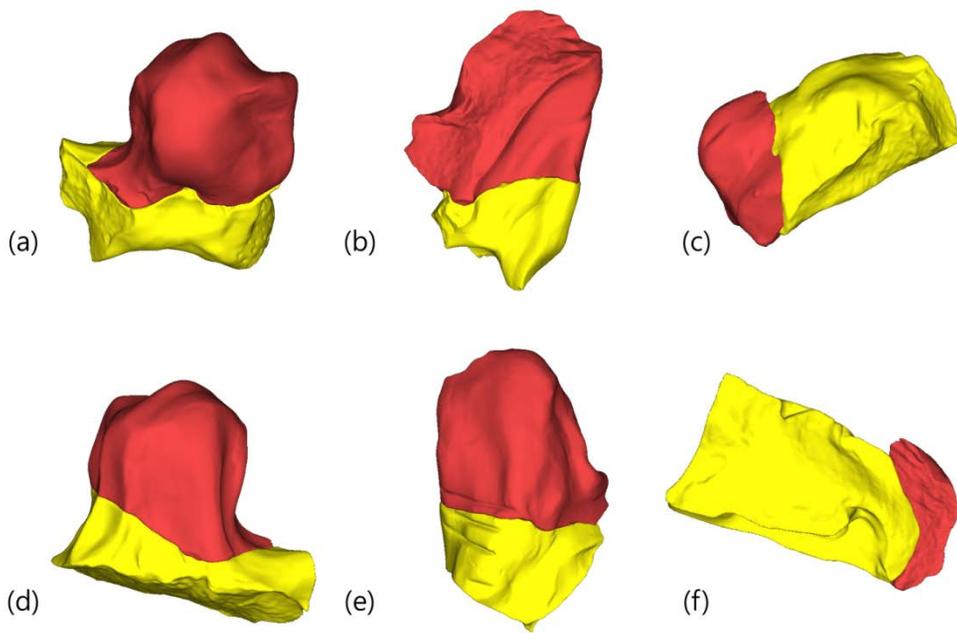


Fig. 5.6 Pairwise matching for sculpture fragments. (b), (e), and (f) are the difference views of (a), (b), and (c), respectively.

The experimental results for the cake models are shown in Fig. 5.5. As shown in Figs. 5.5-(c) and (d), our approach is able to find partial matching properly. Fig. 5.6 shows pairwise matching results of fractured sculpture models in three cases. It is shown that our method matched various fractured surfaces regardless of the size of the fractured surfaces, especially in Experiment 7 (Figs. 5.6-(c) and (f)).

Our approach can be applied to find the global reassembling of several fragments. Every fractured surface of every fragment is extracted first. Through the matching process with feature curves and spin images of feature points, we can obtain similarity measures between two fractured surfaces. From the fractured surface pair with the smallest similarity measure, the transformation matrix is applied to match the fragments unless the two fragments have been matched already by another fractured surface pair. A global reassembling result is obtained by repeating this process until all fragments are matched. Fig. 5.7 shows the global reassembling results of the brick, cake, and sculpture models. These are not precise reassembling results, but there is no doubt that our approach can provide reasonable initial alignments of broken objects in simple ways.

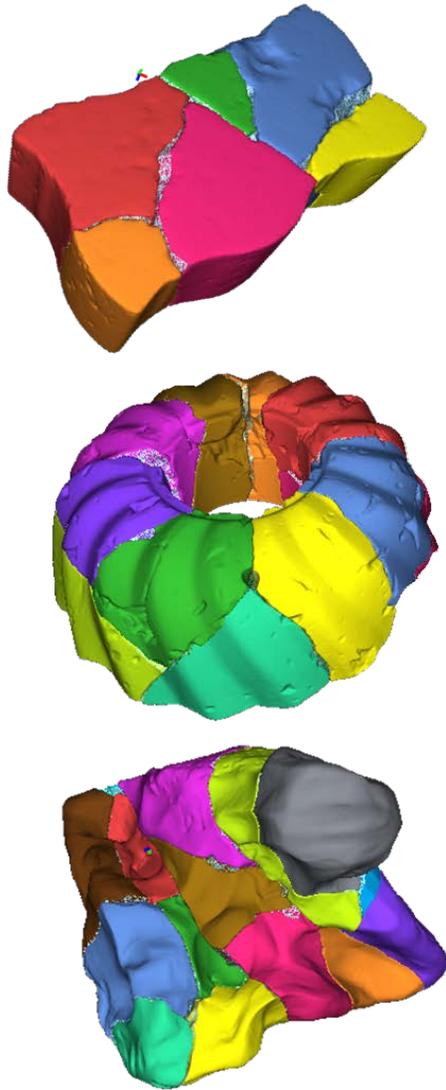


Fig. 5.7 Global reassembling results of brick (top), cake (middle), and sculpture (bottom) models.

## **CHAPTER 6.**

### **ASSESSMENT**

In this study, we use two methods to generate geometry-based virtual fractured surfaces and to use real objects for the quantitative evaluation of reassembly.

#### **6.1 Assessment with virtual fractured surfaces**

##### **6.1.1 Randomly fractured surfaces**

We use deformation based on a least-squares mesh to create a virtual fractured surface [22]. The least-squares mesh is a method of transforming a mesh into a sparse matrix based on mesh connectivity and solving the matrix under a given condition to perform mesh representation or deformation.

There are several conditions for creating a virtual fractured surface. The first is the position and number of deformation seed points. The number of seed points at which deformation occurs on any surface affects the complexity of the generated fractured surface (Fig. 6.1). If the number of deformation seed points is too large, a noisy surface is generated rather than a fractured surface. If the number of deformation seed points is too small, a surface with no distinct feature is generated. In this study, the number of seed points was determined randomly within 2–3% of the total number of vertices of the surface by experiments. The positions of the

deformation seed points were randomly determined on the target surface. However, the boundary region was excluded to avoid abnormal deformation.

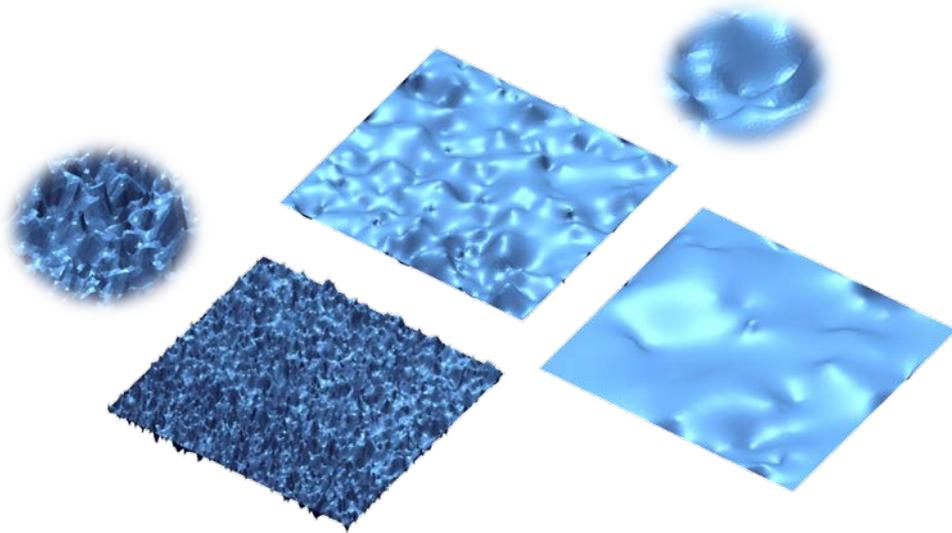


Fig. 6.1 Effects of number of deformation seed points

The second condition is the deformation direction and scale to determine the shape of the fractured surface. The deformation direction is a factor of which direction to move the selected seed point relative to the normal direction. By setting this randomly, complex fractured surfaces can be reproduced. The deformation scale represents the moving distance of the seed point, and its value is proportional to the square root of the surface area. Since the mesh resolution calculated by the average edge length varies depending on the mesh density, the difference between the fractured surfaces generated by the density increases for the same size surface.

Therefore, we choose a scale value that is proportional to the area, not the mesh resolution. Fig. 6.2 shows an example of a fractured surface created by randomly modifying the above factors.

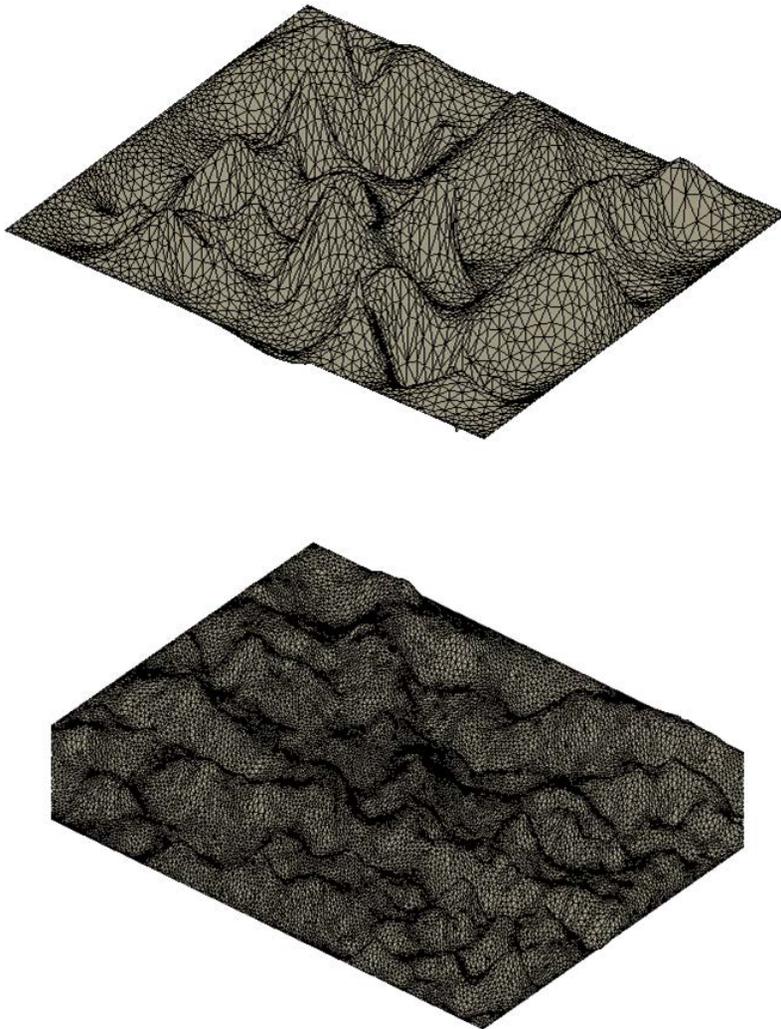


Fig. 6.2 Examples of randomly fractured surfaces.

## 6.1.2 Reassembly with randomly fractured surfaces

The previously generated fractured surfaces are applied to box-shaped objects for reassembly assessment (Fig. 6.3). Deformations of the same scale are generated in opposite directions on the surface where two adjacent fragments meet, and randomly fractured surfaces are generated in the same way on other adjacent surfaces.

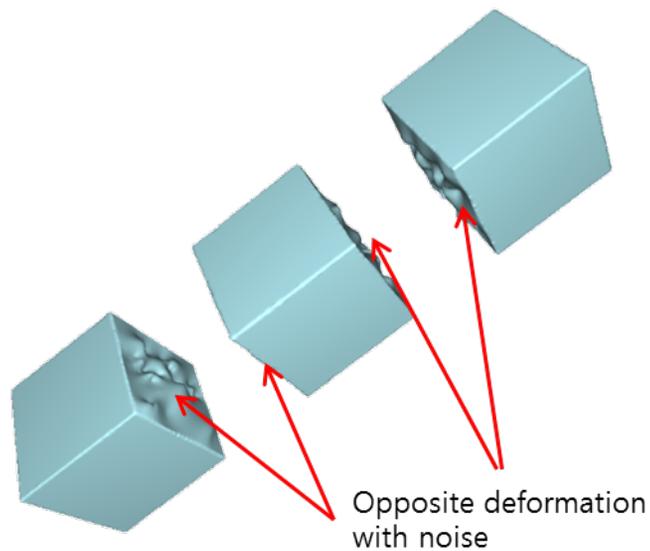


Fig. 6.3 Simple reassembly examples with randomly fractured surfaces.

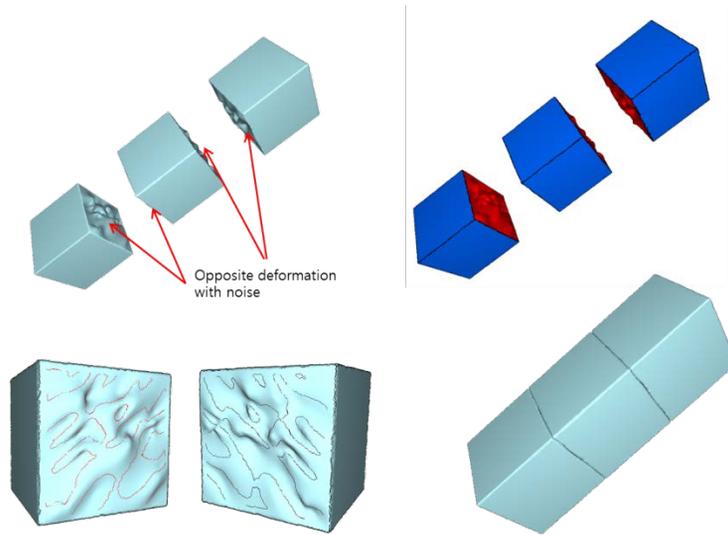


Fig. 6.4 Reassembly process with randomly fractured surfaces

The generated fragments are reassembled in the same manner as explained in CHAPTER 4, and finally, the reassembled result is obtained (Fig. 6.4).

### 6.1.3 Assessment of reassembly

The assessment using the example in Fig. 6.3 is very simple. Since the shape of the fragment is a cube with known size, we can evaluate the reassembly by creating a mesh of three fragments that are ideally assembled and comparing it with a reassembled fragment. However, in this case, it is necessary to remove the fractured surface used in the matching process in the reassembly process. This is because the fractured surface is a surface that does not exist in the original model. That is, it is generated when the fracture occurs (Fig. 6.5).

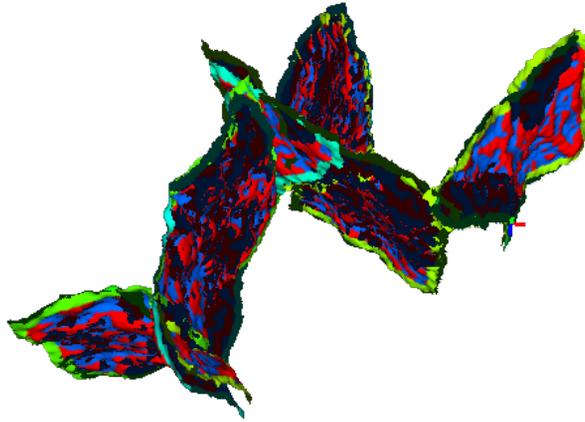


Fig. 6.5 Fractured surfaces after reassembly.

Fractured surfaces can be extracted using the positional relationship of segments after reassembly. Surface segments are classified in the reassembly process. However, this classification is only based on the roughness of the segments. Noisy intact surfaces can be classified as fractured surfaces. In the case of inner fragments (which have no intact surfaces (Fig. 6.6)), actual fractured surface segments are classified as intact surfaces by the clustering algorithm. Therefore, we cannot use the classification information directly.

After the reassembly, the fractured surface has a fractured surface of the other fragment that matches it on the opposite side. The surfaces touching each other are opposite in the normal direction, and the distance is very close. Since the shape of a surface segment can be changed by defects and noise during fracture, a fractured

surface segment can be extracted by using the ratio of vertices that are in contact with other fragments among the vertices included in the segment.

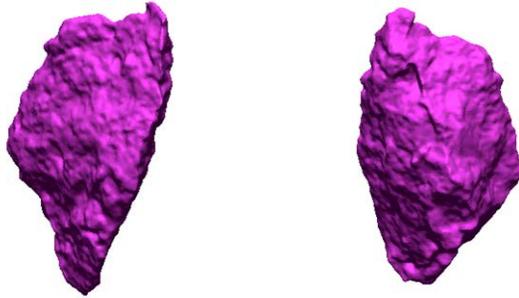


Fig. 6.6 Inner fragment. All surface segments are fractured surfaces.

Fig. 6.7 shows the outer point clouds except the fractured surface in the reassembled box example.



Fig. 6.7 Outer point clouds after reassembly of box example.

Reassembly assessment is calculated using the difference between the original object and the outer surface of the reassembled fragments, as described above. At this time, the original object and the reassembled model are aligned. Fig. 6.8 and Fig. 6.9 show the mesh deviation in a color map and histogram of the box example, respectively.

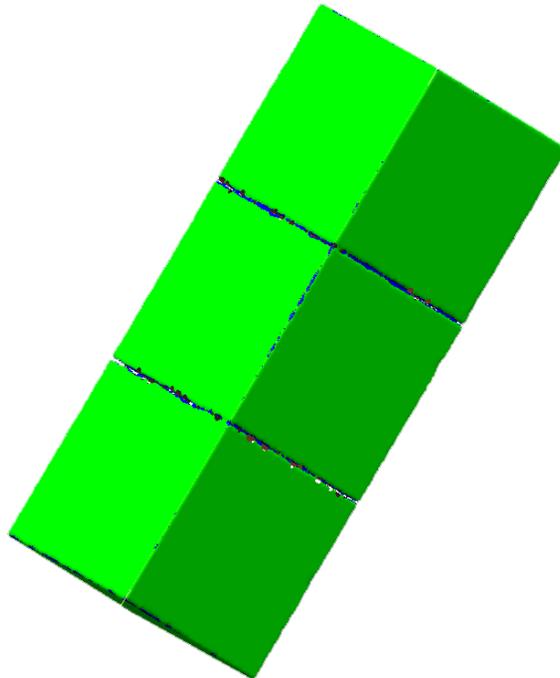


Fig. 6.8 Mesh deviation of reassembled box objects.

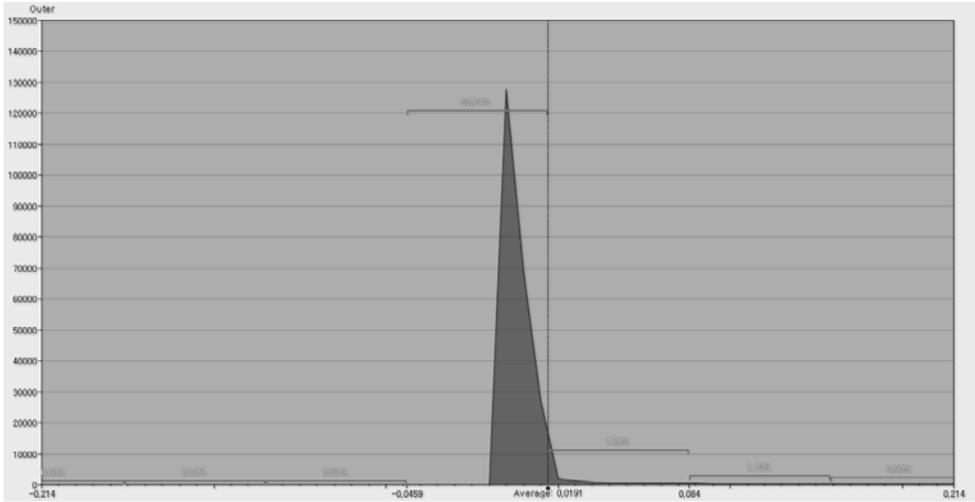


Fig. 6.9 Histogram of mesh deviation of box examples.

In Fig. 6.8, the green area belongs within the allowed deviation, and the allowed deviation is set to the mesh resolution ( $= 0.41007\text{mm}$ ). The mean difference between the original model and the reassembled model is  $0.0191\text{mm}$ , and 99% of vertices are within the allowable deviation.

The following Table 6.1 shows the numeric results of the multiple reassembly performed on randomly generated fractured surfaces. As in the previous example, we created a randomly fractured surface on the adjacent faces of three boxes, added different noise, and then performed a reassembly.

Table 6.1 Reassembly experiment results with randomly generated fractured surfaces.

Test No	Mesh resol.	Fractured Surface 1		Fractured Surface 2		Mesh deviation		
		deform(%)	noise(%)	deform(%)	noise(%)	average	std	Max.
0	0.575552	3.62%	12.83%	3.20%	8.00%	0.01817364	0.01484487	0.0785979
1	0.579557	4.98%	7.72%	4.17%	7.12%	0.02330406	0.01630112	0.0792964
2	0.579093	4.04%	18.25%	4.65%	11.21%	0.01879155	0.01575573	0.0819643
3	0.578518	2.98%	10.48%	4.17%	11.15%	0.01247545	0.00881338	0.0652861
4	0.57721	4.98%	12.38%	3.56%	11.04%	0.01373552	0.00976484	0.0525781
5	0.575527	2.77%	15.61%	4.33%	15.56%	0.02921816	0.02301333	0.116495
6	0.57828	3.41%	12.91%	3.99%	6.83%	0.0091266	0.00994406	0.101833
7	0.578166	3.06%	13.87%	3.14%	15.05%	0.01514516	0.01204015	0.101833
8	0.578901	4.57%	18.07%	3.30%	7.04%	0.02278358	0.01849086	0.101833
9	0.576122	2.60%	17.59%	2.20%	8.03%	0.01650276	0.01319105	0.0709277

Deform (%) represents the degree of deformation of the randomly generated fractured surface. It is calculated as the ratio of the number of vertices on the fractured surface to the number of deformation seed vertices used for deformation. Noise (%) describes the degree of noise added to each fractured surface, and it is the ratio of the number of deformation seed vertices to the number of vertices added with noise.

As shown in the table, several reassembled models had a mean deviation of 0.0179 mm compared to the original model and had a standard deviation of 0.014 mm. Considering the resolution of the used mesh (average 0.57769 mm), we can derive very accurate reassembly results for randomly fractured surfaces.

In this study, we evaluated the robustness of our reassembly technique using the surface signature through a fragment model with virtual fractured surfaces. For two box models sharing an arbitrary fractured surface, Gaussian noise was added to each fractured surface before calculating the surface signature, and this was

followed by reassembly. Gaussian noise was applied to all points on a fractured surface by applying a random value that followed a Gaussian distribution to change the position of a vertex. Sigma values were set to 0.0, 0.25, 0.5, 0.75, 1.0, and 1.5 times the mesh resolution, and the effect of sigma values on Gaussian noise is shown in Fig. 6.10.

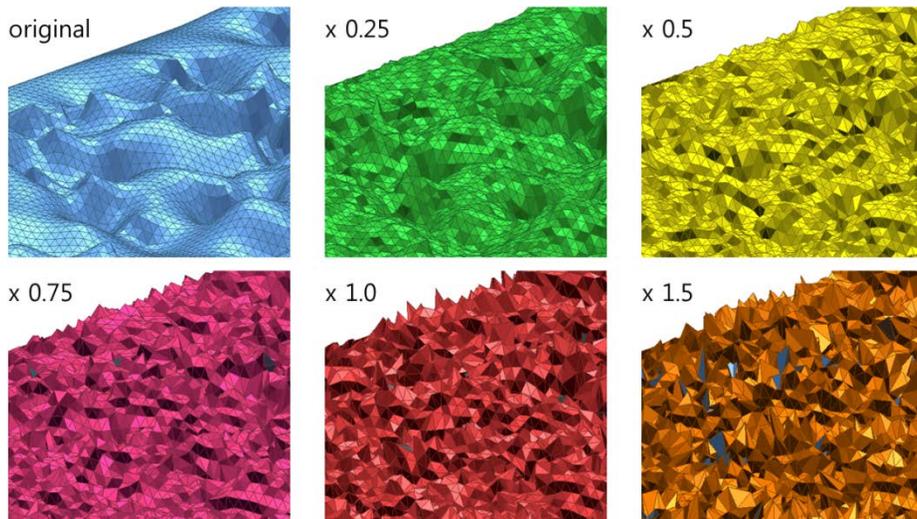


Fig. 6.10 Effects of sigma value for Gaussian noise on the fractured surface.

Table 6.2 and Figs. 6.11 and 6.12 show reassembly results using fractured surfaces with various Gaussian noises.

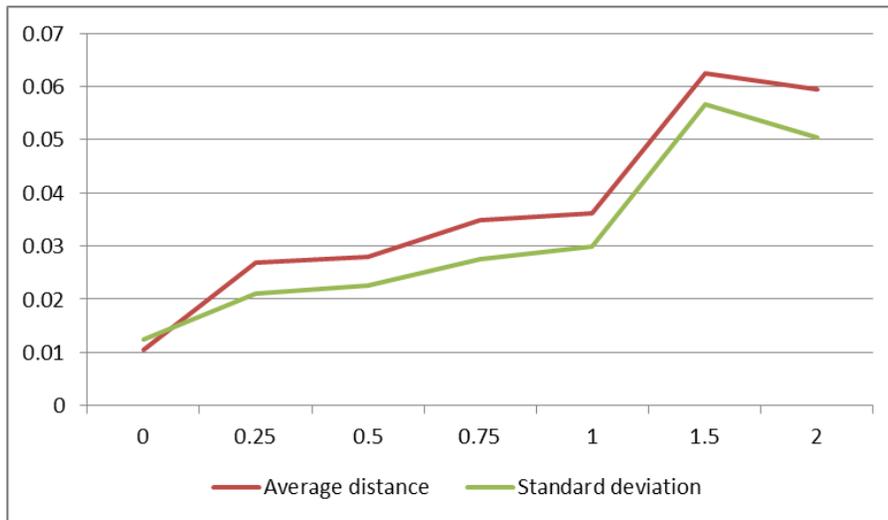


Fig. 6.11 Difference between original model and reassembled model.

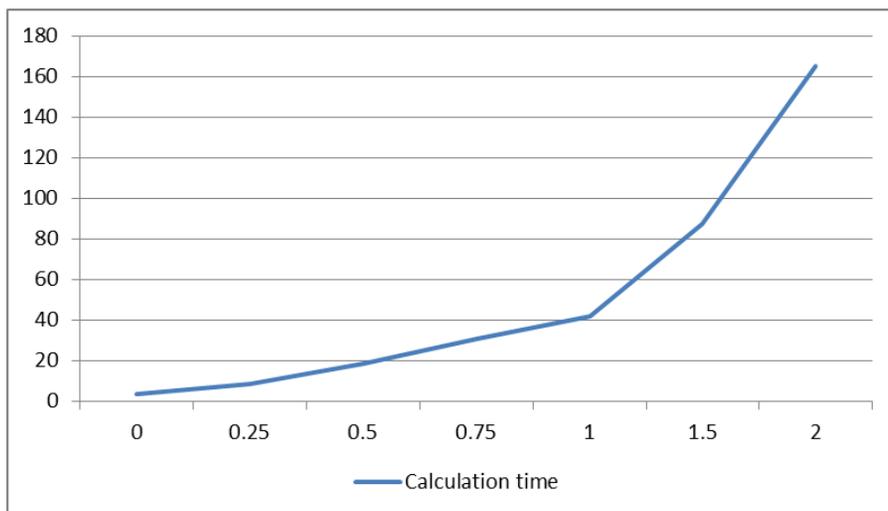


Fig. 6.12 Calculation times for reassembly with varying Gaussian noise.

Table 6.2 Reassembly results with varying Gaussian noise.

Sigma(times of resolution)	0	0.25	0.5	0.75	1	1.5	2
Average distance	0.010484	0.0269008	0.028005	0.0348315	0.0362188	0.0625847	0.0595532
Standard deviation	0.0123672	0.0209801	0.0225589	0.0274966	0.0298307	0.0567399	0.0504225
Max distance	0.0595389	0.124646	0.106939	0.152612	0.172125	0.284268	0.253649
Calculation time	3.738	8.545	18.503	30.865	42.174	87.689	164.822

As can be seen from the tables and graphs, as the noise of the fractured surface increases, the reassembly result becomes more different from the original model (fragment alignment is not accurate for strong Gaussian noise, as shown in Fig. 6.13), and the calculation for reassembling transform time is also lengthened. However, when we look for a plausible reassembly, which is not very accurate on a fractured surface to which Gaussian noise is added so that it is difficult to find the original shape, the proposed surface signature does not depend on local geometry. It can be concluded that it is robust against small noise.

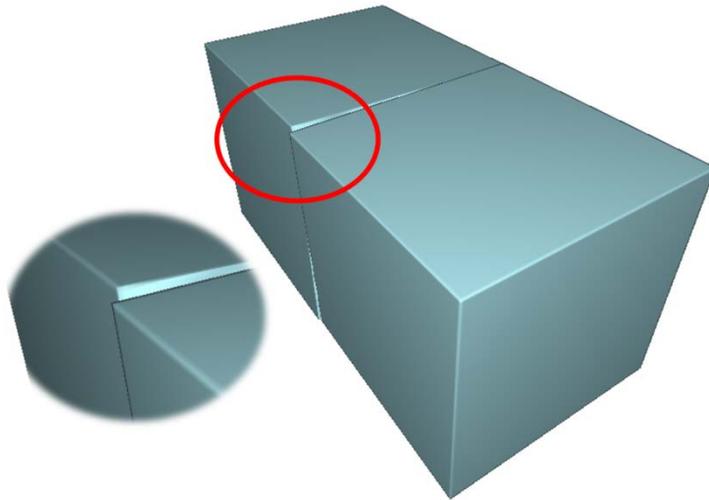


Fig. 6.13 Misaligned fragments with strong Gaussian noise.

## 6.2 Assessment with real broken objects

In this study, a reassembly evaluation using real 3D objects was performed in addition to the reassembly evaluation using virtual fractured surfaces. Fig. 6.15 shows the process of acquiring the mask model and data used in the assessment. The original gypsum model was first extracted as a 3D digitized model through a 3D scanner (DAVID system, 0.06mm precision), and then fragments obtained by generating fractures in the gypsum model were converted into digitized models through a 3D scanner.

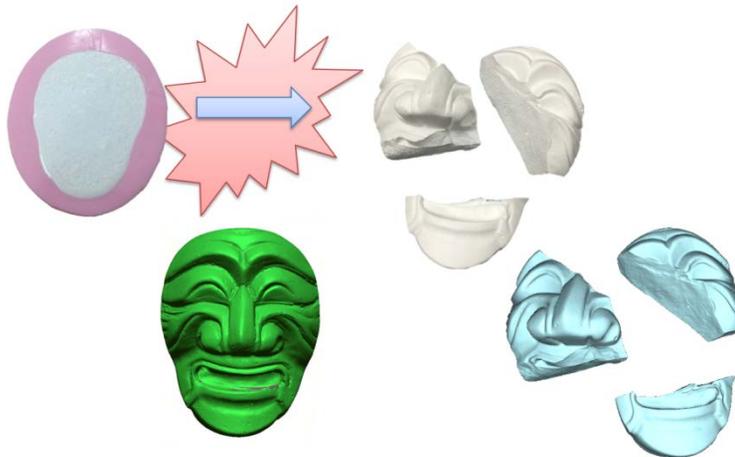


Fig. 6.14 Original model and fragments of gypsum mask model.

The fragment models thus obtained were applied to the reassembly process and reassembled, as shown in Fig. 6.15. The assessment through the comparison of the original model with the outer point clouds in Fig. 6.15 is shown in Fig. 6.16. The

average distance is  $-0.1284$  mm, which is smaller than the mesh resolution of  $0.5343$  mm. However, it can be seen that there is a slight difference compared to the box example due to the broken pieces of the actual object, fragments that have fallen off, and errors occurring in the 3D scanning process.

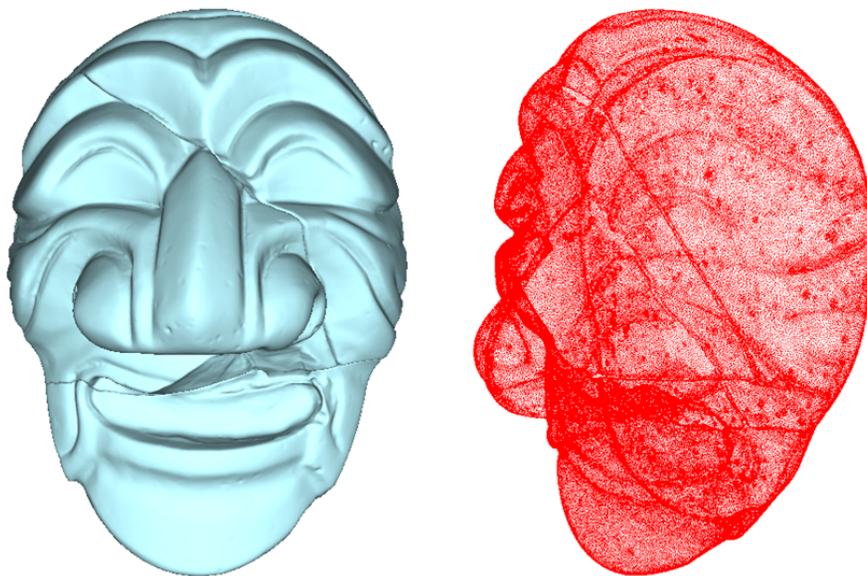


Fig. 6.15 Reassembled mask fragments and its outer point clouds.

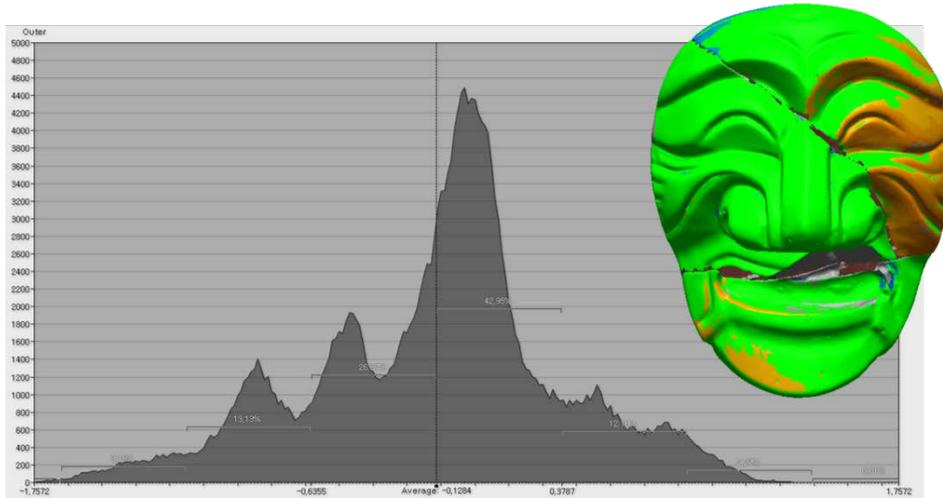


Fig. 6.16 Assessment of mask model reassembly.

### 6.3 Discussion

In this study, we evaluated the reassembly method using virtual fractured objects and real objects. In addition, the accuracy and robustness to the noise of our algorithm were verified by several experiments. It was found that the proposed reassembly process for the various virtual fractured surface geometries yields meaningful results and that the reassembly is performed without any problems, even for the actual model, which is relatively difficult to control.

## **CHAPTER 7.**

### **CONCLUSION**

In this paper, we have proposed a novel descriptor, the surface signature, for reassembling broken objects. In addition, we have proposed a reassembling process based on the surface signature and found out that the surface signature can be applied to the reassembling process. The contributions of our research are as follows. First, a new simple descriptor is introduced for fractured surface matching. This descriptor simply reflects the convex and concave region distribution of the fractured surface. Surface matching is conducted based on the region distribution and results in a plausible reassembly.

During the reassembly process, the complex 3D surface geometry has reduced dimensions in the order of signature distribution, signature curves, and feature points. Simplified information alone yields meaningful results. In addition, we proposed a method to consider the complex fractured surface shape by modifying the definition of segmentation or roughness included in the reassembly process.

Finally, we proposed a quantitative evaluation method using virtual fractured surfaces and demonstrated the effectiveness of the proposed surface signature and matching process. At the same time, we confirmed the robust nature of the surface signature to noise.

At the beginning of this study, we asked whether a fractured surface with complex

shapes could be represented simply and intuitively. By applying our method to various examples and experimenting with virtual fractured surfaces, we have shown that the surface signatures can be applied to reassembly expressing the characteristics of a sufficiently broken object. We hope to extend it to other areas based on geometry shape characteristics in future.

## REFERENCES

- [1] D. Koller, M. Levoy, Computer-aided reconstruction and new matches in the forma urbis romae, *Bullettino Della Commissione Archeologica Comunale di Roma* 2.
- [2] Papaioannou, E.-A. Karabassi, T. Theoharis, Virtual archaeologist: Assembling the past, *Computer Graphics and Applications*, IEEE 21 (2) (2001) 53–59.
- [3] F. Kleber, R. Sablatnig, A survey of techniques for document and archaeology artefact reconstruction, in: *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, IEEE, 2009, pp. 1061–1065.
- [4] A. Tabrum, Reassembly of broken tertiary mammal specimens from pieces collected more than 20 years apart, in: *JOURNAL OF VERTEBRATE PALEONTOLOGY*, Vol. 26, SOC VERTEBRATE PALEONTOLOGY 60 REVERE DR, STE 500, NORTHBROOK, IL 60062 USA, 2006, pp. 130A–130A.
- [5] O. Ron, L. Joskowicz, C. Milgrom, A. Simkin, Computer-based periaxial rotation measurement for aligning fractured femur fragments from ct: A feasibility study, *Computer Aided Surgery* 7 (6) (2002) 332–341.

- [6] A. Willis, D. Anderson, T. Thomas, T. Brown, J. L. Marsh, 3d reconstruction of highly fragmented bone fractures, in: Medical Imaging, International Society for Optics and Photonics, 2007, pp. 65121P–65121P.
- [7] H. Freeman, L. Garder, Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition, *Electronic Computers, IEEE Transactions on* (2) (1964) 118–127.
- [8] D. Goldberg, C. Malon, M. Bern, A global approach to automatic solution of jigsaw puzzles, in: Proceedings of the eighteenth annual symposium on Computational geometry, ACM, 2002, pp. 82–87.
- [9] K. Hori, M. Imai, T. Ogasawara, Joint detection for potsherds of broken earthenware, in: Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on., Vol. 2, IEEE, 1999.
- [10] G. U' c, oluk, I. H. Toroslu, Automatic reconstruction of broken 3-d surface objects, *Computers & Graphics* 23 (4) (1999) 573–582.
- [11] W. Kong, B. B. Kimia, On solving 2d and 3d puzzles using curve matching, in: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, Vol. 2, IEEE, 2001, pp. II–583.

- [12] A. Willis, D. B. Cooper, Alignment of multiple non-overlapping axially symmetric 3d datasets, in: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, Vol. 4, IEEE, 2004, pp. 96–99.
- [13] G. Papaioannou, E.-A. Karabassi, On the automatic assemblage of arbitrary broken solid artefacts, *Image and Vision Computing* 21 (5) (2003) 401–412.
- [14] Q.-X. Huang, S. Flory, N. Gelfand, M. Hofer, H. Pottmann, Reassembling fractured objects by geometric matching, *ACM Transactions on Graphics (TOG)* 25 (3) (2006) 569–578.
- [15] S. Winkelbach, F. M. Wahl, Pairwise matching of 3d fragments using cluster trees, *International Journal of Computer Vision* 78 (1) (2008) 1–13.
- [16] E. Altantsetseg, K. Matsuyama, K. Konno, Pairwise matching of 3d fragments using fast fourier transform, *The Visual Computer* 30 (6-8) (2014) 929–938.
- [17] CHEN, Zhili, et al. Physics-inspired adaptive fracture refinement. *ACM Transactions on Graphics (TOG)*, 2014, 33.4: 113.
- [18] GREGOR, Robert, et al. Automatic 3D Object Fracturing for Evaluation of Partial Retrieval and Object Restoration Tasks-Benchmark and Application to 3D

Cultural Heritage Data. In: 3DOR. 2015. p. 7-14.

[19] SIFAKIS, Eftychios; DER, Kevin G.; FEDKIW, Ronald. Arbitrary cutting of deformable tetrahedralized objects. In: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation. Eurographics Association, 2007. p. 73-80.

[20] HELLRUNG, Jeffrey, et al. Geometric fracture modeling in Bolt. In: SIGGRAPH 2009: Talks. ACM, 2009. p. 7.

[21] ZHU, Yufeng; BRIDSON, Robert; GREIF, Chen. Simulating rigid body fracture with surface meshes. ACM Transactions on Graphics (TOG), 2015, 34.4: 150.

[22] SORKINE, Olga; COHEN-OR, Daniel. Least-squares meshes. In: Shape Modeling Applications, 2004. Proceedings. IEEE, 2004. p. 191-199.

[23] Katz, S., & Tal, A. (2003). Hierarchical mesh decomposition using fuzzy clustering and cuts (Vol. 22, No. 3, pp. 954-961). ACM.

[24] Katz, S., Leifman, G., & Tal, A. (2005). Mesh segmentation using feature point and core extraction. The Visual Computer, 21(8-10), 649-658.

- [25] Golovinskiy, A., & Funkhouser, T. (2008, December). Randomized cuts for 3D mesh analysis. In *ACM transactions on graphics (TOG)* (Vol. 27, No. 5, p. 145). ACM.
- [26] Lai, Y. K., Hu, S. M., Martin, R. R., & Rosin, P. L. (2009). Rapid and effective segmentation of 3D models using random walks. *Computer Aided Geometric Design*, 26(6), 665-679.
- [27] Kalogerakis, E., Hertzmann, A., & Singh, K. (2010). Learning 3D mesh segmentation and labeling. *ACM Transactions on Graphics (TOG)*, 29(4), 102.
- [28] Au, O. K. C., Zheng, Y., Chen, M., Xu, P., & Tai, C. L. (2012). Mesh segmentation with concavity-aware fields. *IEEE Transactions on Visualization and Computer Graphics*, 18(7), 1125-1134.
- [29] Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2), 115.
- [30] Petrelli, A., & Di Stefano, L. (2011, November). On the repeatability of the local reference frame for partial shape matching. In *2011 International Conference on Computer Vision* (pp. 2244-2251). IEEE.

- [31] C. S. Chua and R. Jarvis. Point signatures: A new representation for 3d object recognition. *IJCV*, 25(1):63–85, 1997.
- [32] Mian, A., Bennamoun, M., & Owens, R. (2010). On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *International Journal of Computer Vision*, 89(2-3), 348-361.
- [33] J. Novatnack and K. Nishino. Scale-dependent/invariant local 3d shape descriptors for fully automatic registration of multiple sets of range images. In *ECCV*, 2008.
- [34] F. Tombari, S. Salti, and L. D. Stefano. Unique signatures of histograms for local surface description. In *ECCV*, volume 6313, pages 356–369, 2010.
- [35] Herrmann, Leonard R. (1976), "Laplacian-isoparametric grid generation scheme", *Journal of the Engineering Mechanics Division*, 102 (5): 749–756.
- [36] Taubin, G. (1995, September). A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (pp. 351-358). ACM.

- [37] M. Desbrun, M. Meyer, P. Schroder, and A. H. Barr, “Anisotropic feature-preserving denoising of height fields and bivariate data,” in *In Graphics Interface*, 2000, pp. 145–152.
- [38] Y. Ohtake, A. G. Belyaev, and I. A. Bogaevski, “Polyhedral surface smoothing with simultaneous mesh regularization,” in *Proc. of the Geometric Modeling and Processing 2000*, 2000, pp. 229–237.
- [39] U. Clarenz, U. Diewald, and M. Rumpf, “Anisotropic geometric diffusion in surface processing,” in *Proc. of IEEE Conference on Visualization '00*, 2000, pp. 397–405.
- [40] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher, “Geometric surface smoothing via anisotropic diffusion of normals,” in *Proceedings of IEEE Conference on Visualization '02*, 2002, pp. 125–132.
- [41] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *ICCV'98*, Jan 1998, pp. 839–846.
- [42] K.-W. Lee and W.-P. Wang, “Feature-preserving mesh denoising via bilateral normal filtering,” in *Proc. of Int'l Conf. on Computer Aided Design and Computer Graphics 2005*, Dec 2005.

- [43] C. C. L. Wang, "Bilateral recovering of sharp edges on feature insensitive sampled meshes," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 4, pp. 629–639, July 2006.
- [44] Lavoué, G. A roughness measure for 3D mesh visual masking. In *Proceedings of the 4th symposium on Applied perception in graphics and visualization* (pp. 57-60). ACM. July 2007.
- [45] KARNI, Z., AND GOTSMAN, C. 2000. Spectral compression of mesh geometry. In *ACM Siggraph*, 279–286.
- [46] DRELIE GELASCA, E., CORSINI, M., AND EBRAHIMI, T. 2005. Objective evaluation of the perceptual quality of 3d watermarking. In *IEEE International Conference on Image Processing*, 241–244.
- [47] WU, J.-H., HU, S.-M., SUN, J.-G., AND TAI, C.-L. 2001. An effective feature-preserving mesh simplification scheme based on face constriction. In *Pacific Graphics*, 12–21.
- [48] CORSINI, M., DRELIE GELASCA, E., AND EBRAHIMI, T. 2005. A multi-scale roughness metric for 3d watermarking quality assessment. In *Workshop on image analysis for multimedia interactive services*.

[49] M. Meyer, M. Desbrun, P. Schröder, A. H. Barr, Discrete differential geometry operators for triangulated 2-manifolds, in: Visualization and mathematics III, Springer, 2003, pp. 35–57.

[50] J. MacQueen, et al., Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol. 1, Oakland, CA, USA., 1967, pp. 281–297.

[51] [https://en.wikipedia.org/wiki/K-means\\_clustering](https://en.wikipedia.org/wiki/K-means_clustering)

[52] D. H. Douglas, T. K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, Cartographica: The International Journal for Geographic Information and Geovisualization 10 (2) (1973) 112–122.

[53] A. E. Johnson, Spin-images: a representation for 3-d surface matching, Ph.D. thesis, Citeseer (1997).

[54] A. E. Johnson, M. Hebert, Surface matching for object recognition in complex three-dimensional scenes, Image and Vision Computing 16 (9) (1998) 635–651.

[55] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM* 24 (6) (1981) 381–395.

## ABSTRACT (Korean)

부러진 3차원 물체의 재조립 기술은 3차원 물체 획득 기술과 컴퓨터의 연산 능력의 발전으로 인해 최근 여러 분야에 적용되고 있다. 오래 전부터 중요한 이슈가 되었던 고고학부터 최근 의료 영상을 기반으로 하는 성형 및 정형외과 분야에서 활발하게 연구되고 있다. 하지만 부러진 물체의 파단면이 가지는 복잡하고 울퉁불퉁한 형상은 물체를 재조립하기 위한 특징을 추출하기 어렵게 만들고 있으며, 고차원의 복잡한 형상 정보 및 서술자를 이용하여 물체의 재조립을 수행하도록 한다. 또한 물체가 부러질 때 발생하는 작은 파편 등으로 인한 파단면의 변형도 이 분야의 어려움 중 하나라고 할 수 있다.

본 연구에서는 부러진 물체의 파단면을 간단하면서도 효과적으로 표현할 수 있는 표면 특성 기술자(Surface signature descriptor)를 제안한다. 이 기술자는 사람이 물체를 조립 방법에서 착안하여 파단면을 볼록한 영역과 오목한 영역으로 구분하고, 이 파단면과 반대되는 형상 패턴을 가지는 파단면을 찾아 물체를 조립한다. 이 기술자는 국소적인 형상 정보에 의존하지 않아 파단면에 존재할 수 있는 형상 잡음에 둔감하고, 기존의 고차원의 형상 정보에 비해 간단하고 쉽게 계산할 수 있는 장점을 가진다. 제안한 표면 특성 기술자를 이용하여 실제 부러진 물체의 재조립을 수행

하는 과정에 있어서 복잡한 파단면의 형상을 처리할 수 있도록 표면 구분(segmentation), 표면 거칠기의 정의, 파단면 간의 유사도 측정 기술 등이 함께 제안되며, 실제 부러진 물체를 이용한 재조립의 구현을 통해 본 연구에서 제안하는 방법의 효용성을 설명한다.

또한 부러지기 이전의 검증 자료의 부재로 인한 재조립 기술의 정량적인 평가가 어려운 문제의 해결 방안으로 가상 파단면 생성을 통한 검증 방법을 제안한다. 가상의 파단면은 몇 가지의 요소에 의해 임의적으로 생성되며, 재조립 알고리즘의 정확성, 효율성, 그리고 형상 잡음에 대한 강인함 등 다양한 측면에서의 평가가 가능하며 본 연구에서 제안한 재조립 방법 역시 이를 통해 평가된다.

본 연구는 3차원 물체의 재조립의 시작인 형상 특성에 대한 기술부터 정량적인 평가 방법까지 전체에 걸쳐 재조립에 적절한 기술들을 고안 및 개선하였다는 데에 그 의의가 있으며, 향후 재조립 기술이 적용되는 타 연구 분야로의 확장 가능성을 가지고 있다.

**주요어:** 부러진 물체의 재조립, 파단면, 표면 특성, 가상 파단면 생성, 재조립 평가

**학번:** 2006-20941