IMEKO International Conference on
Metrology for Archaeology and Cultural Heritage
Lecce, Italy, October 23-25, 2017

Classification and Reconstruction Algorithms for the Archaeological Fragments

Nada A. Rasheed[1], Md Jan Nordin[2]

[1]*University of Babylon, Hillah, 51001, Iraq, nadaar@siswa.ukm.edu.my*
[2]*Universiti Kebangsaan Malaysia (UKM), Selangor, 43600, Malaysia, jan@ukm.edu.my*

***Abstract* – This problem is divided into two subtasks: the classification of archaeological fragments into similar groups and reconstruction each group into the original objects. To solve this problem, a method has been proposed, which exploits the color and texture properties of the surfaces of the fragments. Furthermore, the reconstruction of archaeological fragments in 3D geometry is an important problem in pattern recognition. Therefore, this research has implemented the algorithms to reconstruct real datasets using Neural Networks. The challenge of this work is to reconstruct the objects without previous knowledge about the part that should start the assembly; this greatly helps to avoid the presence of gaps created due to missing artifact fragments. The study utilized the geometric features of the fragments as important features to reconstruct the objects by classifying their fragments using a Neural Network model.**

I. INTRODUCTION

Potteries are considered the most common materials used by the human for several thousands of years [1]. Therefore, Archaeology is the science which studies the last remnants of human civilization [2], which means exploring the lives of ancient peoples by examining their remains. The artifacts are often found at archaeological sites in a fractured state and the process of manually reconstruct to unknown objects from a large number of smaller pieces may involve thousands of irregular fragments. It is difficult and expensive process. Additionally time consuming may take years of a tedious work, especially in case of the losing some pieces, or require laborious effort and experienced archaeologists [2][3]. Therefore, the aim of this work is to propose a novel algorithm for classifying fragments depending on global and local features, specifically color and texture features. Moreover, to design a robust prototype for the reconstruction of 3D objects, despite the existence of the gaps, by exploiting the geometric features (especially the slope of the edges of the fragments); as well as finding the appropriate location for matching. Finally, to propose a new method for aligning and matching two different edges of pair fragments, and finding the corresponding point set that is important for matching all kinds of fragments. It is worth mentioning, that the algorithms are implemented in MATLAB application, and performed by laptop computer (Intel Core i5 2.5GHz with 8GB RAM).

II. MATERIALS AND METHODS

This paper presents a proposed framework to solve the problem of classification and reconstruction of archaeological fragments. Therefore, the proposed methodology consists of two phases; each one performs a specific job, as shown in Fig. 1

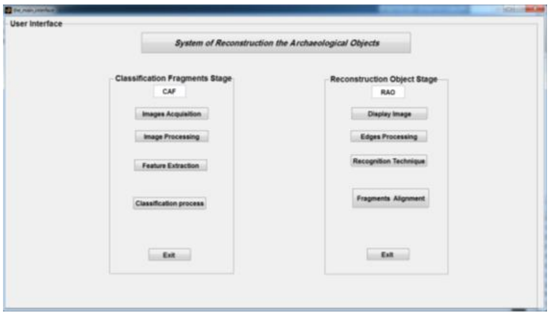

Fig. 1. Represents the interface of the system.

A. Classification Fragments into Groups

This framework consists of a set of procedures, which can be summarized in the following steps: Image acquisition, Object segmentation, Feature extraction depending on color and texture of the surfaces of fragments and Classification the fragments into groups; If the fragment doesn't belong to the group, it should be brought back to the dataset to re input it until being suitable for grouping with other fragments. In the case all fragments formed group, it is transferred to the next stage, which is reconstruction the group into one object.

When the system is running, the user interface window is opened; it contains a set of command buttons. The first one for *Images Acquisition*, which opens a new window directly for the purpose of loading six images of fragments files into memory. Then the fragments are selected by the user from the menu pop-up, as shown in Fig. 2 a.

a *b*

Fig. 2.a. Upload images to the memory.
b. Represents Image Segmentation

In order to obtain the fragments without a background, proposed algorithm has been applied.
Step1: Separate color matrix of image into three matrices (Red, Green, Blue).
Step2: Calculate the mean and standard deviation for the rows of each matrix.
Step3: Fit a Normal Distribution to the data.
Step4: For each matrix, calculate the Threshold value by computing the 95% Confidence Interval for the distribution parameters.
Step 4.1: Compare each pixel with the Threshold.
Step 4.2: If the value of a pixel is less than or equal Threshold, then it changes to 255.
Step 4.3: Else, the pixel has the same value.
Step5: Performs algorithm 4.1 of the three matrices (Red, Green, Blue), where each output pixel contains the median value in 5-by-5 neighborhood around the corresponding pixel in the input image.
Step6: Regroup the three matrices into one matrix.

This algorithm will begin to work by simply pressing the button *Image Processing* that is visible in the main window of the system as depicted in Fig. 1b. It can move between the images by dragging the slider, for the purpose of obtaining the object without background. In order to obtain the features for each fragment, this work depends on the color by calculating the colors intersection for each pair of fragments. Therefore, to extract the RGB color, we rely on the mathematical method that is similar to the one used in [4], which includes the intersection of RGB matrices between each image with the corresponding of the other images. As shown in Fig. 3, this turns out the results of classification before applying the classification procedure. The graph shows the first image achieves the highest value of colors intersecting (8110, 8411) with the third and fifth images respectively, so the three images will be the first group. Also the second image achieves highest intersection (2724, 1783) with the fourth and sixth images respectively; also these images will represent the second group.

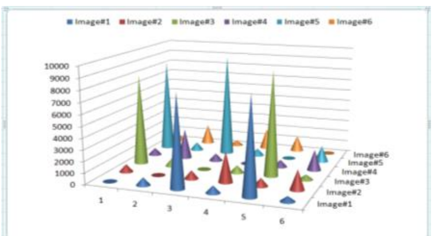

Fig. 3. The intersection between six color images

Many researchers consider the texture of objects or regions is an important feature used to recognize [5]. Thus, this work exploited this feature by utilizing the Local Binary Patterns (LBP) [6]. This method was suggested by Ojala et al. in 1996 [6] to describe the texture of image as a vector that labels the pixels of an image into histogram describing small-scale appearance of the image, for using it for further image analysis [7].

So, the result of the first fragment it is as shown in Fig. 4.

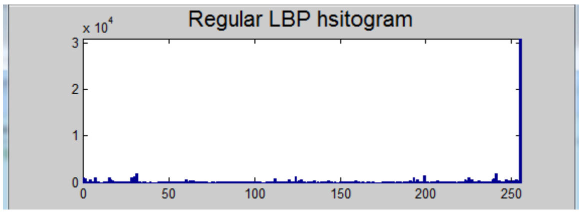

Fig. 4. Histograms of LBP for the first fragment.

The classification operation is divided into two steps, the first step classifies the fragments depending on the color using the proposed algorithm [4], suppose we have two fragments that shown in Fig. 5.

Fig. 5. Two parts of a broken object.

Step1: R= [Color feature for first image,......, Color feature for sixth image].
Step2: Sort Ascending order for color feature of first image *A*.
Step3: Compare the max value in the first column feature of image *A* with intersect values of image *B*.
IF the value of B is the highest value
Group and saved A with B.
else if the value of B>A
Compare the max value of the B with max value of the image that achieved a highest intersect C:
IF max value of the C; Do not group A with B
else; Group A with B and saved the result.
Step3.1: Moved to the second highest value, which represents the intersection with the other image.
Step3.2: Repeat this procedure.
Step3.3: Return to the first step until all the values in the first column finished.
Step4: Repeat the procedure until all columns represents rest the images are complete.

The second step classifies the fragments based on the texture using the Euclidean distance. Thus, the final result will be when the results of classification of color and texture are identical as shown in Fig. 6. Therefore, the results should be saved for the next step, in this example we saved the second and third groups only.

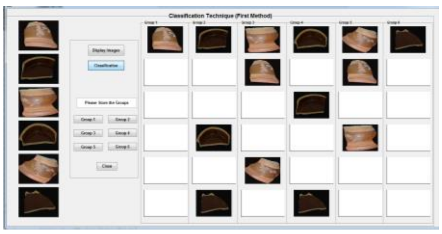

Fig. 6. Result of the classification technique.

B. Reconstruct Each Group into 3D Object

Using Three-Dimensional measurements to resolve the problem of the reconstruction of archaeological artifacts has become widespread [8]. Occasionally, archaeological workers suffer when trying to reassemble each group of ceramic into the object with high accuracy, especially when a presence of significant gaps in the object, or even in the case the fragments of object are so many. The main theme of this part is to propose a system for reconstruction of ancient 3D objects to the original form. The challenge of this work is to assemble the objects without previous knowledge for the part that must start in the compilation and this helps a lot to avoid the presence of gaps during losing parts of the artifacts.

For the purpose of reconstructing the pottery fragments, this work propos method that consists of four major phases; it involves 3D model acquisition by the scanner device, and apply preprocessing procedure to remove the noise that appear as result of scan object; Then geometric features that calculated after extracting the contour of each fragment and dividing it into sub-blocks. Furthermore, recognition the part that will matching pair of fragments by using Backpropagation Neural Network algorithm. Finally, aligning the candidate 3D fragments and matching through proposed new method. If the object will not complete, should be brought back to the dataset, else the procedure was finished, and the object was obtained a fully. See Fig. 7.

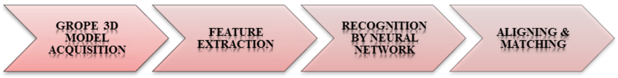

Fig. 7. The diagram of the proposed method to reconstruct the 3D object.

- Acquisition of 3D Model

In order to reconstruct the real datasets, which are consists of two groups, each one comprising of 3 fragments. As shown in Fig. 8

Fig. 8. Illustration Two groups of 3D fragments

Hence, single group will be selected in reconstruction window, which represent one vessel. Therefore, in this stage all the 3D fragments belong to this group will be loading to the memory. As shown in Fig. 9.

Fig. 9. Upload 3D models of fragments.

- Feature Extraction

Considered as the 3D features are better than 2D, when using it for recognition, but it may be more expensive. Axiomatic, when the archaeologists attempt to re-assemble the broken pieces, the assembly will be through the edges of the fragments; that means, extract the geometric features. After that, they will check the texture and the color of the surfaces of the fragments. Thus, in order to extract the features, this paper utilized the edges of the fragments as an important feature to reconstruct the objects, so the following algorithm represents this procedure and the results as shown in Fig. 10:

Step 1: Find all edges in the mesh, note that the internal edges are repeated.

Step 2: Determine uniqueness of edges.

Step 3: Determine the counts for each unique edge.

Step 4: Extract edges that only occurred once.

Step 5: Plot the edges.

Fig. 10. Boundaries of 3 fragments

The aim of this paper is to locate the correct location for matching two fragments and to continue matching the rest of the parts, as shown in Fig. 11. It shows the similarity at the beginning of the slopes of the sub-contours *A* and *B*, and continues with the rest of sub-contours.

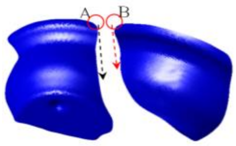

Fig. 11. Illustration Recognize the matching place

In order to locate the correct position, we decided to divide the contour of the fragment to four-parts of equal size as much as possible, and handle each part as a separate object, so for this fragment each part consists of 75, 75, 75, and 72 points. As shown in Fig. 12.

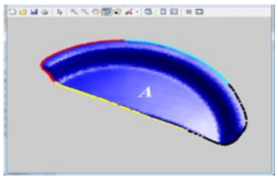

Fig. 12. Divides contour into four equal-sized

Then every part is divided to the sub-contours, each one consists of five points, so in this case we have 59 sub-contours for current fragment and we neglected two remaining points. Therefore, the following algorithm implements this activity.

Step1: Divided the contour into four parts do:

$$fragmentsize = round(size(fragment, 1)/4) \tag{1}$$

$$\begin{aligned} &\text{for } i = 1:4 \\ &splitted\_data\{ii\} = FA((ii-1)\times Nsubsize + (1:Nsubsize),:) \\ &end \end{aligned} \tag{2}$$

Step2: Plot each part

Step3: Divided each part into sub contour (five points)

$$A = cell2mat(splitted\_data(1)); \%part-1 \tag{3}$$

$$\begin{aligned} &N = length(A); \\ &\text{n} = 5; \end{aligned} \tag{4}$$

$$M = fix(N/n)\times n; \tag{5}$$

Step4: Calculate the total number of sub contour

$$b = M/n; \tag{6}$$

$$FS = A(1:M); \tag{7}$$

$$FS\_1 = reshape(A, [5 \ \text{b}]) \tag{8}$$

Considered measure the slope of each sub contour is the main feature to distinguish the place that should be matching pair of different parts for two different fragments. Therefore, we applied the algorithm [9] of 3D slope on each part and the result as demonstrate in Fig. 13.

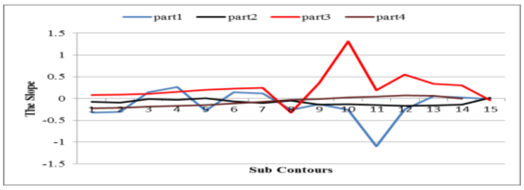

Fig. 13. The slopes for four parts of fragment

Other features extracted in this work based on the all coordinates of each point (*x*, *y*, *z*) to calculate the minimum point for each sub-contour on the axes *x*, *y*, *z* and the same manner we are calculated the maximum, mean, and the variance values between the points of sub contour for four parts. Results are shown in Fig. 14- 17.

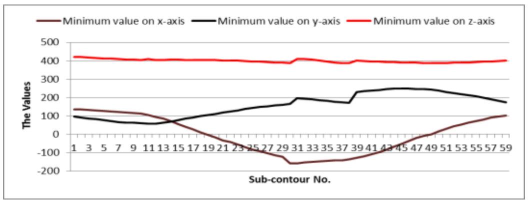

Fig. 14. Minimum Values for Each sub-contour

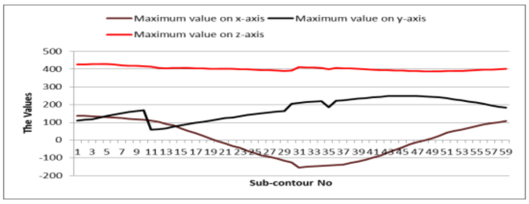

Fig. 15. Maximum Values for Each sub-contour

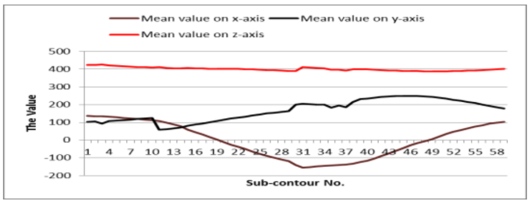

Fig. 16. Mean Values for Each sub-contour

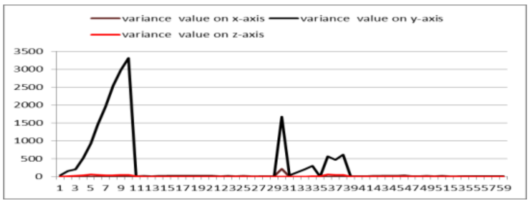

Fig. 17. Variance Values for Each sub-contour

Thus, we will have thirteen features for each sub-contour can possibly use for candidacy a pair of fragments for matching. For the other two pieces as shown in Fig. 18, we see that fragment (*B*) consists of 55 sub-contours with three points being neglected, while the fragment (*C*) consists of 59 sub-contours with four points being neglected. Therefore, after extracting feature vectors we will have a matrix (13×114) ready to be the input of the network. Note that before training a neural network, we must normalize the input.

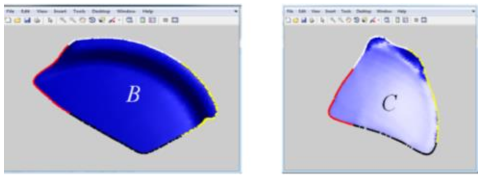

Fig. 18. Two Fragments That Divided into Four Parts

The next step, a procedure has been applied to detect the joint between the pair of sub-contours according to the similarity features, this work used the Backpropagation Neural Network algorithm, which is a powerful mapping network that has been applied successfully to a wide variety of problems [10]. Hence, the structure of neural network composed of three layers, the first layer consists of 13 nodes of input. The second layer namely the Hidden layer that consists of 30 nodes found through the experience. Finally, the target output layer consists of 8 units, because this layer was assigned one node for each part of the fragments (*B* and *C*), so the test data will be in the range of (1,-1) [11]. In order to increase the network effectiveness and make it more suitable, we used the learning rate (0.05) with momentum term (0.9). After training the neural network, we would like to point out that it is completed in 335 iterations of 1000 epochs that we assumed, as shown in Fig. 19.

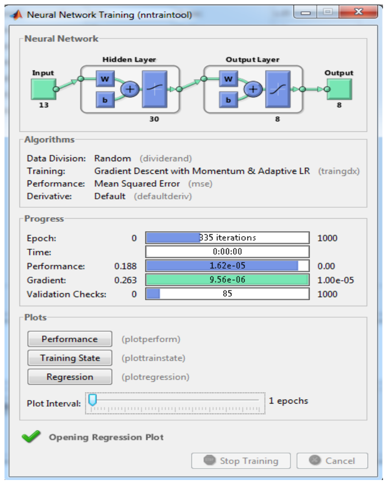

Fig. 19. Neural Network Training

As shown in Fig. 20, in order to recognize the unknown part of the fragment (*A*) to match it with the corresponding parts of the other fragments, we used the features of each part that one we mentioned it in Figures 14-17 separately to be unknown part into the input layer of the network and test it through the Feedforward phase.

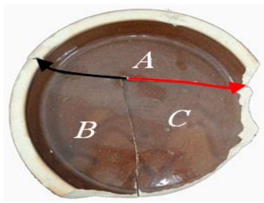

Fig. 20. Represent Broken vessel

After testing each part of the fragment (*A*) the results are shown in Figures 21-24.

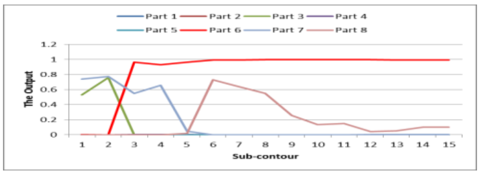

Fig. 21. The test to the Part 1 of the Fragmen A

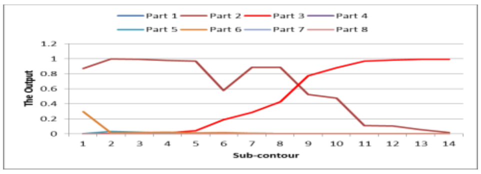

Fig. 22. The test to the Part 2 of the Fragmen A

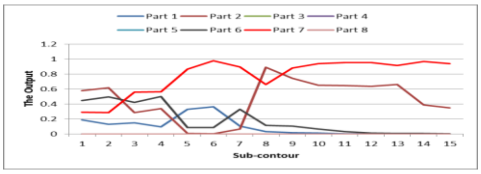

Fig. 23. The test to the Part 3 of the Fragmen A

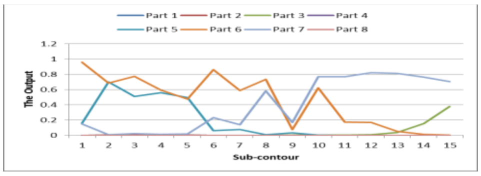

Fig. 24. The test to the Part 4 of the Fragmen A

Subsequently, after computing the actual output; the part of the fragment is a winner if the maximum value between the computed output nodes is close to one. Hence, we can see that the first part of the fragment *A* can match with corresponding of the sixth part of the second fragment, because the red line achieved the highest values between eight nodes, so the assembly begins from the third sub-contour and continues to the end of part. Here, we adopt the maximum value between eight nodes because we assign 1 to the current part and zeros otherwise. Immediately after recognizing the pairs of sub-contour between the edges of fragments *A* and *B* that probably represent the same point in space should be starting to align and match. Often this is done simply by matching each point with its closest neighbor of the other cloud. In this case the angle of rotation, and the distance of transition between the two fragments must be computed, example of this as it is shown in Fig. 25.

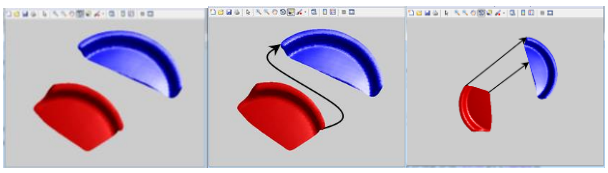

Fig. 25. Two fragments should rotate and transform

Thus, this research suggests one algorithm for matching two edges of different parts of fragments. Suppose we have two fragments (*A*) and (*B*), fragment (*A*) should be fixed and fragment (*B*) should be followed. So in order to centralization the fragment (*A*) data at the zero, follow the following steps.

Step 1: Find the center of fragment *A*:

$$\mathrm{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}; \tag{9}$$

Where the *xyz* data of fragment (*A*)

$$\mathrm{Centroid}_A = \frac{1}{N}\sum_{i}^{N} p_N^i; \tag{10}$$

Step 2: Accumulating matrix (*H*)

$$\mathrm{H} = \sum_{i=1}^{N} (\mathrm{p}_A^i - \mathrm{Centroid}_A); \tag{11}$$

Step 3: Using Singular Value Decomposition (SVD) to find the direction of most variance, and rotate the data to make it the *x*- axis as follows

$$[\mathrm{U,S,V}] = svd(H,0); \tag{12}$$

Where *U* is $m \times n$ and column orthogonal (its columns are eigenvectors of *AAT*); *V* is $n \times n$ and orthogonal; *D* is $n \times n$ diagonal (non-negative real values called singular values). If *H* is $m \times n$ and *m* greater than *n*, then *SVD* computes only the first *n* columns of *U* and *S* is $n \times n$.

$$R = V \times U^T; \tag{13}$$

V is the direction of the most variance

Step 4: Slide the data up the *x*- axis so all the points are $x >= 0$.

$$T = -R \times \mathrm{Centroid}_A; \tag{14}$$

After fixing fragment (*a*) to the origin points, should find the optimal rotation (matrix *R*) and the translation *T* to the fragment (*b*).

Apply the experiments with 3D alignment method are a difficult task for many authors [12], because we need to identify the angle that should rotate the fragment surface (*b*) according to the *x*, *y*, and *z*-axes. Therefore, in order to achieve the best fit alignment, we applied the Dot Product [13] to find the angle on which the fragment (*b*) must be rotated. In this case, the angle was returned inverse cosine ($\cos^{-1}$) of the elements of θ in degrees. Therefore, we can rotate the object around *x*, *y*, and *z*-axes [12], through our experiments, we can achieve the best result when rotate the fragment around the *z*- axis according to the following formula.

$$Rz = [\cos(\theta)\ {-\sin(\theta)}\ 0;\ \sin(\theta)\ \cos(\theta)\ 0;\ 0\ 0\ 1] \tag{15}$$

In order to provide a solution for rotation the object, it will be done according to the equation (16).

$$C = P_b \times R_z; \tag{16}$$

Where *C* is the new 3D point cloud, P_B is the oldest 3D point cloud, and R_z is the rotation matrices. As shown in Fig. 26 the best rotation is around *z*-axis, so this research depended on this type. The other issue is to add a difference distance between the two objects' coordinates for the purpose of transfer (*T*) the object and matching the two sides of the pair various objects.

Fig. 26. Fragments b rotate to the fragment a

One of the fragments should be apply a transformational algorithm in order to bring it close to each other and to achieve the best fit alignment. Therefore, in order to provide a solution for *R* and *T* as in the equation 17, suppose *b* is the 3D point cloud data of the followed fragment:

$$C = P_b \times R_z + T; \tag{17}$$

Where $T = (x + t_x), (y + t_y), (z + t_z)$, and t represents difference distance between the two objects; *R* and *T* are rotate and transforms has been applied to 3D point cloud (*b*) to align it with 3D point cloud (*a*), as best as possible.

Finally, in order to obtain the optimal matching, applied the Euclidean distance formula between the coordinate of each point of edge of fragment (*b*) and all coordinate of points of corresponding fragment and choose the shortest distance. Given two points (*a*) and (*b*), the Euclidean distance is:

$$d(a,b) = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2 + (a_z - b_z)^2} \tag{18}$$

Given a point *b*, and set of points *A*, the Euclidean distance is:

$$d(b,A) = \min_{i \in 1,\ldots,n} d(b,a_i); \tag{19}$$

Where b, a_i indicates to the values of the points that represent two sub-contours which has been classified (*A*) and (*B*), *n* is the vector size. Last step slide the data up the *z*- axis, so all the points will be positive. The algorithm has been tested on several models of fragments, and it has achieved high-precision results in reconstructing the objects to the original forms accurately, even in the cases when pieces were missing. As shown in Fig. 27.

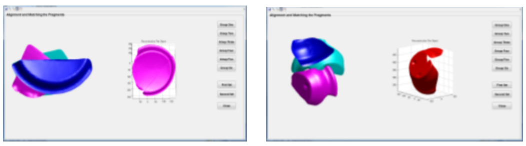

Fig. 27. Illustration Two groups of 3D fragments

III. CONCLUSIONS

For classifying archaeological ceramic fragments, the proposed methods adopted are the color and texture of the surface information of fragments. Moreover, this work has used new feature based on the intersection of global points of the images, and has demonstrated its efficiency to classify the fragments. Also, the effectiveness of the local binary patterns feature has proven to be more flexible with global features, but requires more complex calculations. In addition, this work proposed an approach to reconstruct broken 3D ceramic objects. Therefore, to avoid the gaps that appear due to missing fragments, this work found the slope feature to be suitable for determining the best positions of matching pairs of fragments. When evaluating the first method for accuracy, it achieves value of 96.1% for classification of the fragments into similar groups. Moreover, the geometric features especially the slope of the sub-contour with the neural network algorithm have achieved a good accuracy to reconstruct original 3D objects.

IV. ACKNOWLEDGEMENT

The researchers wish to thank Ministry of Higher Education, Malaysia and Universiti Kebangsaan Malaysia (UKM) for their support on this work via research grants FRGS/1/2014/ICT07/UKM/02/2 and DIP-2016-018.

REFERENCES

[1] H. Mara, “Pottery Plotted by Laser - 3D Acquisition for Documentation and Analysis of Symmetry of Ancient Ceramics,” Springer, New Technologies for Archaeology. Multidisciplinary Investigations in Palpa and Nasca. Peru. Chapter 22, 2012, p. 379.

[2] K. Son, E. B. Almeida, D. B. Cooper, “Axially symmetric 3D pots configuration system using axis of symmetry and break curve,” IEEE Conference on CVision and Pattern Recognition, 2013, pp. 257-264.

[3] L. D. Angelo, P. D. Stefano, C. Pane, “Automatic dimensional characterization of pottery,” Elsevier, Journal of Cultural Heritage, 2017, pp.1-11.

[4] N. A. Rasheed, M. Jan Nordin, “Archaeological Fragments Classification Based on RGB Color and Texture Features,” Journal of Theoretical and Applied Information Technology, vol.76, No.3, June 2015, pp.358-365.

[5] R. M. Haralick, K. Shanmugan, I. Dinstein, “Texture features for image classification,” IEEE Trans. Systems, Man Cybern vol. 3, 1973, pp. 610-621.

[6] T. Ojala, M. Pietikainen and D. Harwood, “A comparative study of texture measures with classification based on feature distributions,” Pattern Recognition, vol. 29, No. 1, 1996, pp. 51–59.

[7] M. Pietikäinen, A. Hadid, G. Zhao, T. Ahonen, “Local Binary Patterns for Still Images, ” Chapter 2 of Computer Vision Using Local Binary Patterns, Computational Imaging and Vision 40, Springer, Verlag London, 2011, pp.13-47.

[8] S. Zheng, Y. Zhou, R. Huang, L. Zhou, X. Xu, C. Wang, “A Method of 3D measurement and reconstruction for cultural relics in museums,” International Archives of The Photogrammetry, Remote Sensing and Spatial Information Sciences, Melbourne, Australia, 2012, pp. 145-149.

[9] D. R. Maidment, D. Tarboton, “Computation of slope,” GIS In Water Resources Class, University of Texas At Austin, 2011, pp.1-9.

[10] R. O. Duda, P. E. Hart, D. G. Stork, 2001, “Pattern classification,” Edition: 2nd, Chapter 6, Wiley-Interscience publication: 10-74.

[11] M. H. Beale, M. T. Hagan, H. B. Demuth, “Multilayer neural network architecture. In Neural Network Toolbox User's Guide,” Matlab R2015a. Mathworks, 2015, pp. 2-9.

[12] H. M. Kjer, J. Wilm, “Evaluation of surface registration algorithms for pet motion correction,” Studienarbeit, Technical University of Denmark, Dk-2800 KGS, Lyngby, Denmark, 2010, pp.28-35.

[13] Z. H. Nitecki, “Calculus in 3d geometry, Vectors, and Multivariate Calculus,” Tufts University, 2012, pp.102-114.