Automatic Reassembly of Fragments for Restoration of Heritage Site Structures

Sivapriya V[a], Madhumitha Senthilvel[a] and Koshy Varghese[a]

[a]Department of Civil Engineering, IIT Madras, India
ce13b046@smail.iitm.ac.in, madhusvel@gmail.com, koshy@iitm.ac.in

Abstract –

A major bottleneck activity in the process of restoration of Heritage Structures is the reassembly of its fragments. Computer-aided reassembly could assist in finding the relation between them thereby reducing time, manpower and potential degradation to fragile fragments. Using geometric compatibility between the adjacent fragments as the central idea, a reassembly framework for a three-dimensional shell is proposed as a logical extension of the two-dimensional framework. Edges are extracted as polygons and relevant features are computed at each of its vertices. Sequences of the match for two fragments in the feature space are found using a modified version of Smith-Waterman Algorithm. Each match is assessed using a connectivity score. The final choice of best match is left to the user by displaying the resultant assembled fragments of prospective candidates along with the score. After pairwise matching, the global reassembly is done through a clustering-based method. This framework can handle fragments even with curved edges which can be reasonably approximated by a set of edges. We verify the methodology using a simulated dataset for both 2D pieces and a shattered 3D surface object.

Keywords –

Fragment automatic reassembly; Partial Curve matching; Feature extraction; 3D Reassembly; Heritage Restoration

1 Introduction

A large number of fragments are discovered during archaeological excavations and heritage site restoration tasks. Finding their relative positions is a very laborious yet crucial step. Manual methods are time-consuming, demands the construction of special supporting structures, labor intensive for heavy fragments and can potentially damage the fragile pieces during the process.

With advancements in technological developments to digitalize objects, computers can aid to a great extent in automating the process. [2] talks about the methods and technologies available for three-dimensional digitization of objects and monuments. [3] introduces a 3D digitizing pipeline for cultural heritage where they brief about polygonizing the point cloud to mesh to represent the geometry of the object.

This problem is similar to solving a three-dimensional jigsaw puzzle. There are a variety of approaches to solve the problem. The fragmented pieces themselves contain the clue for solving it. The clue varies from colour compatibility in the case of paintings [4], incisions on the surface for stones, marble veining directions [6], hand impressions on pottery [7] etc. Sometimes the clue also comes from knowing the end result like in the case of Skulls where similarity matching could be performed with the standard template to reassemble [8]. For highly eroded fragments, reassembly can be performed with the objective of maximising its packaging efficiency.

Unlike paintings, unearthed and abandoned fragments of buildings do not have colour or colour variations. Neither do we have a clear idea of the end result of the structure after reassembly of the fragments. The scope of the paper is to assemble pieces which have retained their geometric information without making any assumptions about the reassembled piece. We follow a framework for assembling pieces in two dimensions making use of only its geometry by contour curve matching. This has applications in assembling flat fragments like the fresco, walls. We then, logically extend the framework to assemble a three-dimensional surface object. Applications include reassembly of surfaces like domes, thin shell structures etc. This paper aims to help archaeologists reassemble interactively with user involvement only in the approval stage.

2 Related Works

The problem of reassembly has different approaches and is dealt in different ways depending on its applicability for each scenario. Every problem is essentially three dimensional. But some problems can be converted to two dimensions. Instances like flat fragments where the third dimension is uniform and does not exhibit appreciable variation belong to this

category. Square type jigsaw puzzles involve utilizing boundary pixel values as the parameter to search among other pieces since there is no variation of shape along the edges [10]. Geometric reassembly dates back to 1964 where Freeman et al. encode the boundary as chains and partially matches chainlets through features for a commercially produced jigsaw puzzle [9]. As mentioned in [11], generalized reassembly is different from the jigsaw puzzles because of the uncertainty in the endpoint, unlike jigsaw puzzles where the delimiting point is fixed. The combination of geometrical and colour matching is very suitable to handle fragments of paintings.

A significant project in this domain was The Digital Forma Urbis Romae project where reassembly of marble pieces belonging to a giant map of Rome is attempted [6]. The clue, in this case, is contained in the incisions of the surface. Prediction of the pattern of the adjacent fragment based on the boundary drawings and searching for it through a number of pieces had resulted in significant improvements in its reassembly. Additional constraints come from the nature of the material - marble veining direction. Here we seek to reassemble apictoral fragments which provide us with no other clue than its edge contour. A similar problem was attempted in [12] and [13] where the fragments were converted to invariant features and compared in the feature space. Different features, the way matches are detected and their global reassembly methods are explored in various works.

In case of surface fragments like potteries, the two-dimensional simplifications do not hold. Some authors make use of the symmetry of objects by utilizing the thickness profile to reassemble pottery [5]. In this paper, we look at a generic case of reassembling 3D surface object without utilizing any information other than its edge information. Thus the proposed framework can handle missing pieces since it relies only on the strength of edge matching.

The extent of automation also varies for different works. A few works require the user to specify initial constraints like iso-planarity, adjacency while others do not involve the user at all and perform a completely automatic reassembly [12] [14]. We use very low user intervention in the following processes: (1) to finalize the pairwise and (2) reject the match when overlaps occur during global reassembly.

3   Framework

This paper aims to propose a 3D surface object reassembly framework by extending that of a 2D reassembly and verifying both the frameworks using a simulated dataset. The steps involved in the framework are described in the following sub-sections.

3.1   Extracting the Contour

The only piece of information we extract from the fragment is its contour. We attempt an apictoral reassembly. While extracting the boundary, we observed many consecutive 'almost' collinear points in the contour, which created a lot of edge jaggedness. This can be seen in Figure 1. We use Douglas-Peucker Algorithm to remove these points which do not contribute significantly to the matching process [17]. This reduces the processing complexity significantly. The choice of the threshold given as an input to the algorithm should be optimum since smaller values can be less effective and higher values can alter the shape of the fragment itself making it less suitable for the further process.
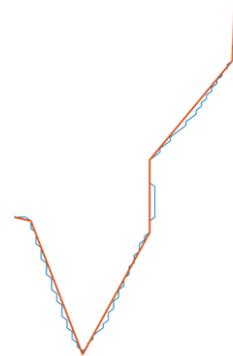

Figure 1. A fragment - Input to the Douglas Peucker algorithm (green) and output (red) with threshold = 0.01

3.2   Feature Extraction and normalization

With this reduced number of points in the contour which better capture the shape variation, we extract the following features based on [12] at every vertex:
I. Log of mean of edge lengths
II. Log of Absolute Value of signed curvature
III. Internal angle
For fragment i, let Fi_c contain the features extracted from the contour clockwise and Fi_a anticlockwise. The perimeter of each fragment is also stored in PM.

Distances in the feature space are highly dependent on the scale of each feature. Thus, to obtain uniformity, we use min-max normalization.

3.3   Pairwise matching

We use a modified version of Smith Waterman algorithm to match the two fragments i and j using Fi_c and Fj_a or Fi_a and Fj_c [15]. Two vertices are said to be same if their Euclidean distance in the feature space is less than Φ.

The endpoint to break the matching sequence is, cyclic matching i.e., we duplicate the sequence and

append it to then end, and then match the sequence by removing repetitions [13]. The output of this is a list of matches between the fragments. The sequences greater than a minimum connectivity value C(i,j) as defined in [12] is stored in G.

$$C(i,j) = \frac{\text{Length of common boundary}}{\text{Minimum } (PM_i, PM_j)} \qquad (1)$$

The transformation in the 2D space is estimated, checked for overlaps after transformation and displayed along with the connectivity value to the user to finalize the match.
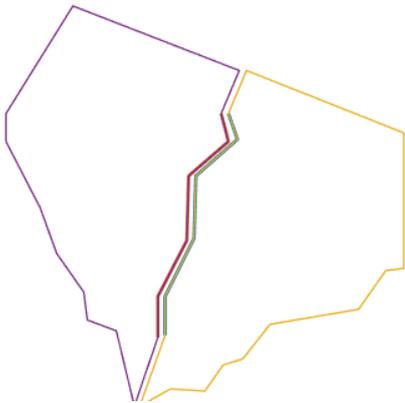

Figure 2. Fragment 2 and 3 pairwise match with Φ=0.05

3.4 Global reassembly

The order of assessment and reassembly of fragments is essential since wrong matches at initial stages potentially lead to error accumulation. Initially, connectivity score is found for all possible pairs of fragments (highest of all the matches for the pair of fragments is chosen). As proposed in [12], we use an agglomerative clustering algorithm. Pairwise matching is continuously performed for the pairs in decreasing order of their connectivity. The scores are not calculated after each iteration significantly reducing the computational requirement. After the reassembly, the fragments merge and form a single piece. Now, if any fragment is left over, the connectivity is recalculated considering the assembled fragment as a single fragment because the newly formed fragment structure might pose a good match to the leftover fragments. For our test dataset, a single iteration is adequate to assemble all of the fragments when we follow the approach of following the decreasing connectivity order. The assembled structure is shown in Figure 3.
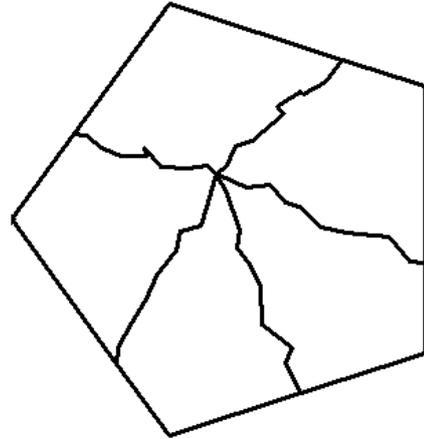

Figure 3. Assembled fragments

4 Extending towards a 3D case

The framework is easily and directly extendible to a 3D scenario. The input model now contains information in three dimensions. Techniques and technologies available to produce the digital representation of solid objects and monuments are discussed in [2]. The kind of features we compute now would differ because of the third dimension.

4.1 Creating the dataset

A unit sphere was created and shattered into 10 pieces using the 3D computer graphics application Autodesk Maya. They were randomly translated along three axes and rotated at different angles. Each individual piece is stored separately and labeled for further processing.
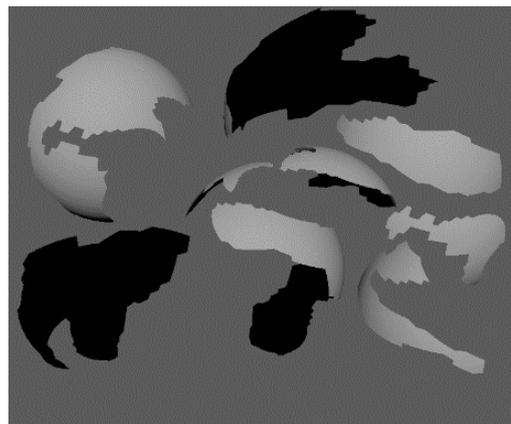

Figure 4. Input fragments

4.2 Extracting the contour

The created fragments were stored as object files (.obj) since it stores the mesh representation of the object as a set vertices with their coordinates, set of edges making up each triangle. Edges which are not common to two triangles indicate the end of the surface and are extracted to form a closed figure. Douglas Peucker algorithm is modified to be applied in three dimensions to smoothen the curve. As mentioned earlier, the threshold value decides the extent of smoothening and is necessary to maintain it constantly for all pieces. For our dataset threshold (th) was chosen as 0.005. The number of vertices constituting the contour drop drastically and hence makes the processing easier.
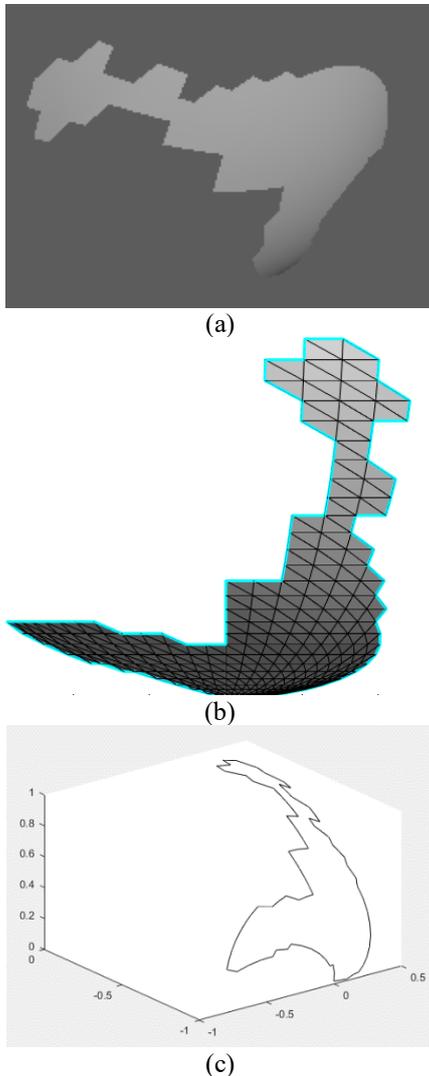
(a)

(b)

(c)

Figure 5. (a) Fragment when generated (b) Its mesh representation and borders highlighted (c) Extracted contour as the output of Douglas Peuckar algorithm with threshold th = 0.005

4.3 Feature Extraction

Features extracted for a 3D polygon at its vertices are different from that of a 2D polygon owing to its non-planar nature. Every curve representing a fragment has a set of points P_1, P_2 … P_n. Each point has the location in the form of 3D coordinates associated with it. We use the numerical approximation used in [18] and [19] for each of the feature defined for a smooth curve.

The features extracted here are:

1. Curvature: It measures the amount of deviation of a curve from a straight line. Mathematically, it can be obtained by calculating the reciprocal of its radius. The numerical approximation for curvature is

$$\alpha_i = 4 \cdot area_i/(a_i \cdot b_i \cdot c_i) \quad (2)$$

where
$area_i = \sqrt{(s_i \cdot (s_i - a_i) \cdot (s_i - b_i) \cdot (s_i - c_i))}$
a_i = Distance between P_i and P_{i-1}
b_i = Distance between P_i and P_{i+1}
c_i = Distance between P_{i-1} and P_{i+1}
$s_i = 0.5 \cdot (a_i + b_i + c_i)$

2. Derivative of Curvature: This measures the rate of change of curvature with respect to arc length. This can be numerically approximated by

$$\beta_i = \frac{3 \cdot (\alpha_{i+1} - \alpha_{i-1})}{2 \cdot a_i + 2 \cdot b_i + d_i + e_i} \quad (3)$$

where
d_i = Distance between P_{i-2} and P_{i-1}
e_i = Distance between P_{i+2} and P_{i+1}

3. Torsion: This is an indicator of the curve's non-planarity by measuring the speed at which the osculating plane rotates along the curve. A numerical approximation is given by

$$\gamma_i = 0.5 \cdot (\gamma'_i + \gamma''_i) \quad (4)$$

where
$\gamma'_i = (6 \cdot H'_i)/(\alpha_i \cdot e_i \cdot f_i \cdot g_i)$
f_i = Distance between P_{i+2} and P_i
g_i = Distance between P_{i+2} and P_{i-1}
$H'_i = 3 \cdot V'_i/area_i$
$$V'_i = \frac{\det(P_{i+2}-P_{i-1}, P_{i+2}-P_i, P_{i+2}-P_{i+1})}{6}$$
$\gamma''_i = ((6 \cdot H''_i))/((\alpha_i \cdot d_i \cdot h_i \cdot j_i))$
$\gamma''_i = (6 \cdot H''_i) / (\alpha_i \cdot d_i \cdot h_i \cdot j_i)$
h_i = Distance between P_{i-2} and P_i
j_i = Distance between P_{i-2} and P_{i+1}
$H''_i = 3 \cdot V''/area_i$
$$V''_i = \frac{\det(P_{i-2}-P_{i+1}, P_{i-2}-P_i, P_{i-2}-P_{i-1})}{6}$$

Here we take the average of γ' and γ'' to avoid introducing asymmetry in our calculations. As [16] states, only the above three metrics are enough to parameterize a Euclidean signature since they are the fundamental signature invariants. Feature extraction is

done both clockwise and anticlockwise.

4.4 Pairwise Matching

We use the modified form of Smith Waterman algorithm like in the previous case except for the features matched. Two points A $(\alpha1,\beta1,\gamma1)$ and B $(\alpha2,\beta2,\gamma2)$ of two different fragments score a similarity score of 1 if the distance between them in the feature space is less than the threshold. For matching fragments i and j, features extracted clockwise for fragment i and counterclockwise for fragment j or vice versa is given as input to the algorithm.

A connectivity score as defined in (1) is evaluated for each pair and values below $\mu=0.1$ are discarded. Once a good match (> 0.1) is found, the appropriate 3D transformation of one of the piece is estimated and made to merge with the second fragment and displayed (as shown in Figure 6) for the user to approve the connection. Once approved, the connectivity score and the start and end vertices of the matching part is stored separately.
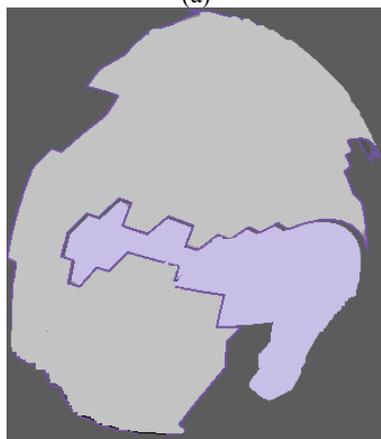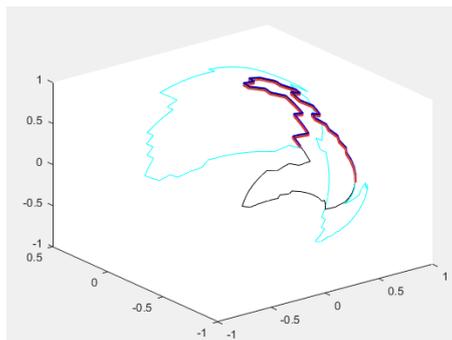

(a)

(b)

Figure 6. Pairwise matching of fragment 2 and 8. (a) shows Fragment 2 (cyan) and Fragment 8 (black) with the matching portions of the fragments in blue colour and red colour respectively. (b) shows the three-dimensional merged fragments after translation

Table 1. Matrix showing the connectivity values of fragments (a) as estimated by the algorithm and (b) as calculated while creating the fragments. The rows and columns indicate fragment number. (c) shows the percentage error in connectivity value for each pair.

	Observed Connectivity values for pairs of fragments								
	1	2	3	4	5	6	7	8	9
1									
2	0.127								
3	0.129	0.138							
4	0	0.163	0						
5	0	0.25	0.209	0					
6	0.313	0	0	0	0.306				
7	0.142	0	0	0.26	0	0.336			
8	0	0.556	0.298	0	0	0	0		
9	0	0	0	0.472	0.253	0	0	0	

(a)

Actual Connectivity values for pairs of fragments								
1	2	3	4	5	6	7	8	9
0.161								
0.288	0.1865							
0	0.2498	0						
0	0.2917	0.2502	0					
0.369	0	0	0	0.377				
0.26	0	0	0.311	0	0.3995			
0	0.6161	0.3435	0	0	0	0		
0	0	0	0.524	0.318	0	0	0	

(b)

	Percentage Error in connectivity values for pairs of fragmnets								
	1	2	3	4	5	6	7	8	9
1									
2	20.97								
3	55.13	26.01							
4	0	34.75	0						
5	0	14.3	16.47	0					
6	15.13	0	0	0	18.88				
7	45.41	0	0	16.37	0	15.89			
8	0	9.755	13.25	0	0	0	0		
9	0	0	0	9.992	20.44	0	0	0	

(c)

The maximum error in the connectivity value was 55% (for fragment 3 with fragment 1) and the minimum was 0% (for multiple fragments as shown in Table 1c). This maximum value occurs when nearly half the original connection was undetected by the algorithm. The minimum value indicates a perfect fit.

4.5 Global reassembly

For each pair of fragments, the connectivity value calculated and approved by the user is recorded in the form of a matrix as shown below. The values less than 0.1 are reported as 0 as shown in Figure 7. The connectivity values obtained for our dataset as shown in

the figure indicate that each fragment has found a strong connection with atleast one other fragment implying if no overlaps detected, just one iteration is enough to assemble back all the nine fragments.

Starting with the pair having the highest connectivity value, each pair is transformed and merged. At any point throughout the global reassembly process, the user can intervene and undo a merge if any sort of wrong overlaps is visually detected.

A major advantage is that the connectivity value need not be calculated over and over again. Since there were no overlaps, in one iteration all of the pieces were assembled back to form the assembled object as shown in Figure 7.
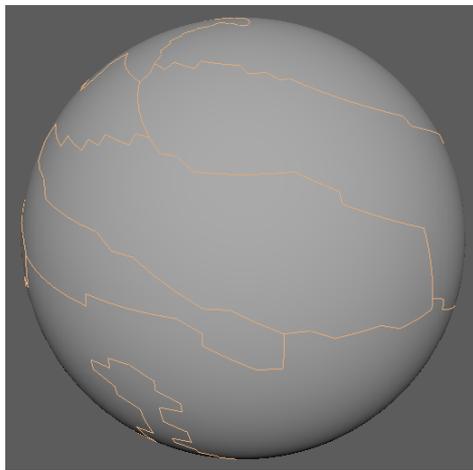

Figure 7 Assembled fragments

5. Results

From table 1c and Figure 7, it can be seen that despite having error values for pairwise connectivity, the sphere was assembled correctly because the location of matches detected were correct though only partially matched. The reasons for a partial match of the common curve could be because of the quality of the image processed and stored as vertices, choice of threshold given as input to Douglas Peucker algorithm or the sampling rate of points along the boundaries of the object.

6. Conclusion and Future work

A framework for re-assembling 3D surface object as an extension of the 2D reassembly framework was proposed and verified using simulated datasets. The merits of this framework are manifold. It is an unsupervised reassembly where the reassembled structure is not known prior to the process. The global reassembly does not compute the connectivity values after each pairwise merging reducing computational

requirement. Additionally, while the user involvement was minimal, this involvement along with the cluster-based merging process prevented error accumulation. For the given dataset, the reassembly was completed in the first iteration. This way, an efficient tool for the archaeologist to speed up the reassembly is put forward.

In future work, the proposed framework will be tested on a real datasets where the fragments are not predefined by the user. This methodology can be used in addition to features extracted from fracture surfaces for validation while reassembling a solid object.

Acknowledgement

References

[1] Willis, A. R., & Cooper, D. B. Computational reconstruction of ancient artifacts. *IEEE Signal Processing Magazine*, *25*(4), 2008.

[2] Pavlidis, G., Koutsoudis, A., Arnaoutoglou, F., Tsioukas, V., & Chamzas, C. Methods for 3D digitization of cultural heritage. *Journal of cultural heritage*, *8*(1), 93-98, 2007.

[3] Li, R., Luo, T., & Zha, H. 3D digitization and its applications in cultural heritage. *Digital Heritage*, 381-388, 2010.

[4] Tsamoura, E., & Pitas, I. Automatic color based reassembly of fragmented images and paintings. *IEEE Transactions on Image Processing*, *19*(3), 680-690, 2010.

[5] Stamatopoulos, M. I., & Anagnostopoulos, C. N. 3D digital reassembling of archaeological ceramic pottery fragments based on their thickness profile. *arXiv preprint arXiv:1601.05824*, 2016.

[6] Koller, D., Trimble, J., Najbjerg, T., Gelfand, N., & Levoy, M., Fragments of the city: Stanford's digital forma urbis romae project. In *Proc. Third Williams Symposium on Classical Architecture, Journal of Roman Archaeology Suppl* Vol. 61, 237-252, 2006.

[7] Kampel, M., H. Mara . Robust 3D reconstruction of archaeological pottery based on concentric circular rills. In *Proc. of the 6th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'05)*, pages 14–20, 2005.

[8] Yin, Z., Wei, L., Li, X., & Manhein, M. An automatic assembly and completion framework for fragmented skulls. In Int'l Conf. on Computer Vision, pages 2532–2539, 2011.

[9] Freeman, H., & Garder, L. Apictorial jigsaw

puzzles: The computer solution of a problem in pattern recognition. *IEEE Transactions on Electronic Computers*, (2), 118-127, 1964.

[10] Alajlan, N. Solving square jigsaw puzzles using dynamic programming and the hungarian procedure. *American Journal of Applied Sciences*, *6*(11), 2009.

[11] McBride, J. C., & Kimia, B. B. Archaeological fragment reconstruction using curve-matching. In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on* (Vol. 1, pp. 3-3). IEEE, 2003.

[12] Lalitha, K. S., Das, S., Menon, A., & Varghese, K. Graph-Based Clustering for Apictorial Jigsaw Puzzles of Hand Shredded Content-less Pages. In *International Conference on Intelligent Human Computer Interaction,* pages 135-147. Springer, Cham, 2016.

[13] Üçoluk, G., & Toroslu, I. H. Automatic reconstruction of broken 3-D surface objects. *Computers & Graphics*, *23*(4): 573-582, 1999.

[14] Palmas, G., Pietroni, N., Cignoni, P., & Scopigno, R. A computer-assisted constraint-based system for assembling fragmented objects. In *Digital Heritage International Congress (DigitalHeritage),* pages 529-536, 2013.

[15] Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. Journal of Molecular Biology 147, 195–197, 1981.

[16] Grim, A., O'Connor, T., Olver, P. J., Shakiban, C., Slechta, R., & Thompson, R. Automatic reassembly of three-dimensional jigsaw puzzles. *International Journal of Image and Graphics*, *16*(02), 1650009, 2016.

[17] Douglas, D. H., & Peucker, T. K. 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), 112-1221, 1973.

[18] Boutin, M., Numerically invariant signature curves, Int. J. Computer Vision 40 pages 235–248, 2000.

[19] Calabi, E., Olver, P.J., Shakiban, C., Tannenbaum, A., and Haker, S., Differential and numerically invariant signature curves applied to object recognition, Int. J. Computer Vision 26 pages 107–135, 1998.