

# DEVELOPMENT OF AN ALGORITHMIC PROCEDURE FOR THE DETECTION OF CONJUGATE FRAGMENTS

D. Filippas, A. Georgopoulos

Laboratory of Photogrammetry, School of Rural & Surveying Eng. National Technical University of Athens  
9, Iroon Polytechniou, 15780 Zografos Athens, Greece  
[filippas9@hotmail.com](mailto:filippas9@hotmail.com), [drag@central.ntua.gr](mailto:drag@central.ntua.gr)

**KEY WORDS:** Conjugate fragments, 3D scanning, linear algorithmic procedure, Cultural Heritage

## ABSTRACT:

The rapid development of Computer Vision has contributed to the widening of the techniques and methods utilized by archaeologists for the digitization and reconstruction of historic objects by automating the matching of fragments, small or large. This paper proposes a novel method for the detection of conjugate fragments, based mainly on their geometry. Subsequently the application of the *Fragmatch* algorithm is presented, with an extensive analysis of both of its parts; the global and the partial matching of surfaces. The method proposed is based on the comparison of vectors and surfaces, performed linearly, for simplicity and speed. A series of simulations have been performed in order to test the limits of the algorithm for the noise and the accuracy of scanning, for the number of scan points, as well as for the wear of the surfaces and the diversity of shapes. Problems that have been encountered during the application of these examples are interpreted and ways of dealing with them are being proposed. In addition a practical application is presented to test the algorithm in real conditions. Finally, the key points of this work are being mentioned, followed by an analysis of the advantages and disadvantages of the proposed *Fragmatch* algorithm along with proposals for future work.

## 1. INTRODUCTION

Over the last few years, archaeological science increasingly utilizes the possibilities offered by the use of computers and especially by the innovations implemented in the field of Computer Vision. A significant reduction in processing time required for the reconstruction of ancient monuments with the application of automatic or semi-automatic methods and the 3D digitization of fragments has been witnessed. Within this framework, this paper focuses on exploring an innovative algorithm, suitable for three-dimensional matching of broken objects. Specifically an algorithmic process, which is based mainly on the 3D geometry of the fragments, is being proposed and developed. In this paper the limits, the reliability and the problems of this process are being examined through simulations and a practical application.

## 2. RELATED PREVIOUS WORK

Reconstructing broken objects from fragments with computer-aided techniques represents a challenge that has been studied for many years (Willis and Cooper 2008). The first approach in this discipline faced the problem of solving jigsaw puzzles (Freeman and Garder 1964, Goldberg et al. 2004). Jigsaw puzzles have a number of important constraints, which aid matching and represent the most basic scenario for automatic reconstruction in 2D space. Contour matching techniques provide solutions to more general problems, without distinguishing between specific edges or features. Applied to 2D space, some of the most outstanding techniques can be found mainly in (Da Gama Leitao and Stolfi, 2002) and in (Papaodysseus, et al., 2002). A very common extension of contour matching techniques in archaeological fragment reconstruction is pottery re-assembly techniques. These are applied to revolution surfaces and can take advantage on the additional constraint of axial symmetry (Kampel and Sablatnig, 2004, Karasik and Smilansky, 2008, Papaodysseus et al., 2002, Ucoluk and Toroslu, 1999). Surface matching techniques are applied to more general problems, considering 3D input data, and provide solutions to

problems with six degrees of freedom (3 translations and 3 rotations). The first approach was presented by Papaioannou and Karabassi (2003), Papaioannou et al. (2002) and Papaioannou et al. (2001) with the underlying assumption that the fractured faces were nearly planar and they matched each other completely. Using a projective space, GPU (Graphics Processing Unit) depth maps were analyzed to reconstruct the original object. An optimization of this technique is further presented by Belenguer and Vidal (2012). Stanford's Digital Forma Urbis Romae project (Koller and Levoy 2006) dealt with heavily eroded fragments, whose fractured surfaces sometimes did not even touched each other. In this approach, instead of using the geometry, reconstruction is done by matching incisions on the fragment's top surfaces.

The technique presented by Huang et al. (2006) reassembles solid objects by first identifying fractured regions and then generating clusters of feature patches for alignment-based matching. This approach also introduces an effective technique for multi piece global matching of fragments. Finally, Brown et al. (2010) exploit the orientation constraints of flat fragments to achieve a simple, fast matcher based on edge geometry. The proposed technique analyzes exhaustively every possible alignment of a pair of fragments in a few seconds.

## 3. FRAGMENT MATCHING ALGORITHM

The idea that spawned the *Fragmatch* algorithm is that the comparison between the broken surfaces should be accomplished by a linear and quick method. That is the main innovation in relation to all previous investigations.

### 3.1 Data input

The algorithm accepts as input data point clouds, i.e. X, Y, and Z for each point of the broken surfaces and not of the whole fragment. It is assumed that while scanning the broken surface should be located in front of the scanner, so that the Z axis defines the depth of the broken surface. The program asks the user to manually enter three pieces of information, i.e. i) the

number of pieces ii) the number of broken surfaces for each piece and iii) the name of the file for each broken surface, which corresponds to the data point cloud.

### 3.2 Comparison based on ICP algorithm

The use of ICP has been employed in order to accelerate the software application. The most common use of the ICP is to match two groups of data points into a common coordinate system (Gelfand et al., 2003). However in this case it is used as a method of comparing the similarity between the broken surfaces. During the application of ICP, two tables (A and B) are being created. The results from the pair wise comparisons of all the surfaces are being placed in table A. The first two columns of table A contain the compared surfaces; the third column takes values 0 or 1 depending on the success of the ICP matching. The fourth column contains the value of the error. Table B contains the compared surfaces, which have been evaluated with an acceptable error.

### 3.3 Creating surfaces and vectors

The point clouds are converted to surfaces through the process of Delaunay triangulation; this is the only non-linear procedure and is necessary for the implementation of *Fragmatch* algorithm. The triangles formed for each broken surface are sorted according to their size, while triangles with sides exceeding a manually set limit are excluded.

After the formation of the surfaces, the normal vector of each triangle is determined. The application point is the centre of gravity of each triangular surface. Each vector's norm is equal to the area of the applied triangle. This characteristic ensures the uniqueness of triangles and vectors.

### 3.4 Global Matching

For each broken surface a table K is being created. The dimensions of this table are  $m \times 7$ ,  $m$  being the number of the created triangles. The first three columns of table K define the coordinates of the triangle's centre of gravity, which is also the normal vector application point. The next three columns define the three vector's components and the last column contains the norm of the vertical vector, which is the area of the triangle. Table K is sorted in descending order of the seventh column. Then the angles between the vectors are being calculated. The calculation of all the angles is being made by the dot product.

| kl <15x7 double> |          |          |         |          |          |          |          |
|------------------|----------|----------|---------|----------|----------|----------|----------|
|                  | 1        | 2        | 3       | 4        | 5        | 6        | 7        |
| 1                | -11.0538 | -7.3775  | 0.0091  | 5.3370   | -21.2573 | 142.7569 | 144.4295 |
| 2                | -1.0478  | -5.3909  | 0.1464  | 2.4778   | -17.4574 | 142.9328 | 144.0163 |
| 3                | 2.8590   | 1.4259   | 0.0859  | -6.3420  | 5.1277   | 84.8327  | 85.2238  |
| 4                | 3.6390   | -7.7407  | -0.4073 | 8.0078   | -14.9049 | 83.1788  | 84.8822  |
| 5                | -5.7475  | -9.3129  | -0.3455 | -2.3262  | -9.3352  | 69.2358  | 69.9010  |
| 6                | 8.7990   | -2.0663  | 0.5976  | -1.7396  | 1.1161   | 51.5858  | 51.6272  |
| 7                | 2.6607   | -12.1780 | -0.4244 | -6.5905  | 14.0927  | 46.7871  | 49.3059  |
| 8                | -14.4499 | -4.3122  | 0.1238  | -7.7506  | 11.6384  | 42.8362  | 45.0607  |
| 9                | -10.8899 | -0.4792  | -0.1690 | -11.2545 | 18.6761  | 34.6692  | 40.9563  |
| 10               | -2.7939  | 0.3776   | 0.2082  | 4.0358   | 20.7774  | 30.6492  | 37.2473  |
| 11               | 7.0778   | -10.2756 | 0.1039  | -11.9505 | 12.0007  | 14.2729  | 22.1483  |
| 12               | 8.0560   | -5.8383  | 0.1210  | -10.2721 | -7.2158  | 15.4097  | 19.8757  |
| 13               | -10.4018 | -1.7214  | 0.3395  | -1.1470  | -2.6634  | 16.3441  | 16.5994  |
| 14               | -2.3059  | -0.8647  | 0.7167  | 0.2270   | -1.3916  | 12.6122  | 12.6907  |
| 15               | -2.8468  | -13.9679 | -0.4564 | -0.2274  | 5.8996   | 8.1835   | 10.0909  |

Figure 1. Table K.

Each angle is being formed by the respective vector with the first vector of the table. The angles are compared based on the order they were registered. Then the subtractions between two angles are being calculated. The final success rate depends on the number of subtractions that were below the *tolerance limit*. The area is calculated according to the seventh column of table K. Two sums are being calculated for each comparison, one

sum for every surface. The matching percentage is estimated based on the following formula:

If  $a_1$  and  $a_2$  are sums of the triangle areas and  $a_1 < a_2$  then

$$Area (\%) = a_1/a_2 * 100 (1) \quad \text{else} \quad Area (\%) = a_2/a_1 * 100 (2)$$

**3.4.1 Angle Tolerance:** The tolerance of the subtraction between two angles is defined by the following formula:

$$\varphi' = \arctan (d/\alpha) (3)$$

$d$  is the surface's wear and  $\alpha$  is the triangle's side. The following figure illustrates two conjugate surfaces; each of them is composed by two triangles. Theoretically the two compared angles are  $\theta_1$  and  $\theta_2$ , however because of the appearing wear in the second surface, the formed angle, between the face 3 and face 4 is the  $\theta_2'$  (Figure 2). The subtraction between angles  $\theta_1$  and  $\theta_2$  is the requested angle  $\varphi'$  (Figure 3).

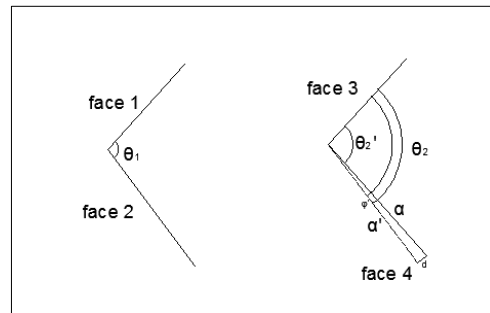


Figure 2. Comparison of angle  $\theta_1$  and angle  $\theta_2$ .

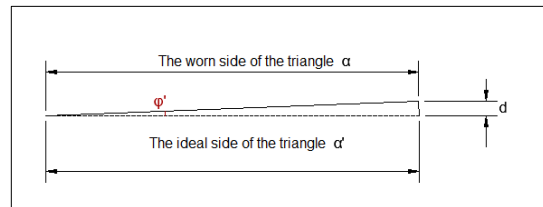


Figure 3. The requested angle  $\varphi'$ .

### 3.5 Partial Matching

In partial matching the same methodology is being applied, i.e. comparison of vector angles and areas. By partial matching the correspondence between fragments of different dimensions is implied. As in the case of global matching, for each broken surface a table K is again created. Assuming that the smaller surface is subsurface of a bigger one, common elements are being identified based on the seventh column of table K. The elements, whose area's subtraction is less than a defined threshold, are considered as common. Then all accepted elements are placed in a new table F. Having two matrices of the same length, matching is being performed in the same way as in the case of global matching. In this case except of the angles' and areas' success rate, a percentage of the greater surface covered by the smaller one, is being determined and displayed.

### 3.6 Steps of the algorithm

The input data include (i) the number of pieces, (ii) the number of broken surfaces of each piece and (iii) the name of the file for each broken surface, which corresponds to the point cloud. The output data is the table with possible conjugate couples and their matching percentages both for the area and angles criterion.

Then the algorithm follows the following procedure:

1. It compares point clouds with ICP algorithm and creates tables A and B.
2. It creates surfaces and vectors for each fragment.
3. It performs the comparison of the pieces.
  - If table B is empty, the pieces are being compared for partial matching. Tables A1 and B1 are being created with the final results.
  - If table B is not empty, its data are being compared for global matching and a table E1 with the final results is being created. Remaining elements are compared for partial matching and matrices A1 and B1 with the final results are being created.

## 4. SIMULATIONS AND EVALUATION

### 4.1 Introduction

The first simulation is going to determine the limits of tolerable noise. The second factor, the accuracy of the scan, is going to be studied with the same methodology as the first factor. Then the next simulation ensures whether the algorithm is affected not only by the number of scanning points, but also by the number of created triangles. One of the most common effects for the surfaces of fragments is corrosion, mainly due to time. Hence, the next simulation studies the performance of the algorithm, handling damaged surfaces. The last simulation analyses whether the algorithm's performance is being affected by the size and shape of the surfaces. The algorithm was created to examine objects whose sides range from 2-3 cm to 1 meter and therefore object dimensions vary between these limits. Comparisons in all simulations were made between two broken surfaces.

### 4.2 Noise and accuracy of scanning

In this simulation the size of the compared surfaces is shown in the following table:

|         |     |         |     |         |     |
|---------|-----|---------|-----|---------|-----|
| Dx (mm) | 280 | Dy (mm) | 280 | Dz (mm) | 163 |
|---------|-----|---------|-----|---------|-----|

Table 1. Size of the surfaces

The algorithm was implemented for three surfaces having the following characteristics (Figs. 5-7):

|   |      |
|---|------|
| <b>Triangles</b>                              | 160  |
| <b>Average triangle area (mm<sup>2</sup>)</b> | 337  |
| <b>Average triangle side (mm)</b>             | 19.7 |

Table 2. Characteristics of the Surfaces.

The algorithm is being performed for three surface pairs. In the first case, the two surfaces are invariant. In the second case the second surface has undergone alterations up to 0.1mm. In the last case the second surface has undergone alterations up to 1mm. The results are presented below:

|                   |          |           |         |
|-------------------|----------|-----------|---------|
| <b>Points</b>     | 100      |           |         |
| <b>Dz (mm)</b>    | 0        | up to 0.1 | up to 1 |
| <b>Area (%)</b>   | 100.0000 | 99.9910   | 99.9597 |
| <b>Angles (%)</b> | 100.0000 | 92.4051   | 60.7362 |

Table 3. Results of the three cases

The results of the first simulation reveal that with a fixed number of triangles, the success rate of the algorithm is reduced, as the distortions increase, which is more or less expected. Since the distortion in the z-axis (depth) increases, different triangles are being created in two different surfaces. However the final result in the above cases is that the three pairs constitute possible conjugate fragments.

### 4.3 Number of scan points

The second part of simulations evaluates the results as far as the number of scan points and the created triangles. The algorithm was applied four times in this experiment. In each application, the input data were two point clouds. The first does not have any distortion, while the second has distortion up to 0.1mm in all three axes x, y and z. The studied subjects had the same dimensions so as to be comparable.

|   |       |       |       |       |
|---|-------|-------|-------|-------|
| <b>Points</b>                                 | 50    | 500   | 5000  | 50000 |
| <b>Dx(mm)</b>                                 | 400   | 400   | 400   | 400   |
| <b>Dy(mm)</b>                                 | 400   | 400   | 400   | 400   |
| <b>Dz(mm)</b>                                 | 280   | 220   | 221   | 240   |
| <b>Average triangle area (mm<sup>2</sup>)</b> | 1265  | 101   | 10.5  | 1     |
| <b>Triangles</b>                              | 78    | 884   | 9055  | 91060 |
| <b>Average triangle side (mm)</b>             | 38    | 10.8  | 3.5   | 1.1   |
| <b>Area(%)</b>                                | 99.98 | 99.99 | 99.94 | 99.85 |
| <b>Angles (%)</b>                             | 94.73 | 26.38 | 10.51 | 25.01 |

Table 4. Results of the four applications of the algorithm

The results (Table 4) indicate that objects of the same size, as the number of points and triangles increases, the percentage of acceptable angles is being decreased. This implies that the algorithm has a greater tolerance when dealing with the area section, compared to angles section. In Figures 4 & 5 the surfaces are visualized and in Figures 6 & 7 the two graphs show the variation of the matching percentages with the variation of the number of scan points.

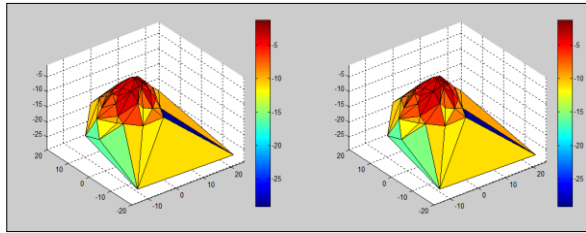


Figure 4: The first pair of surfaces, with 50 points and 78 triangles each.

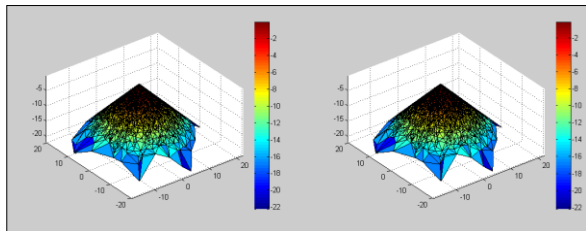


Figure 5: The third pair of surfaces, with 5000 points and 9055 triangles each.

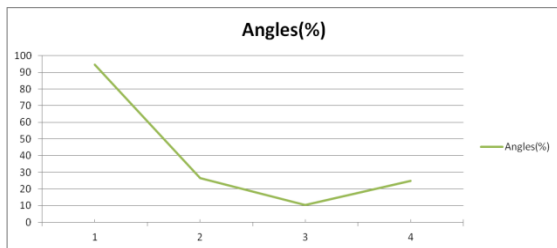


Figure 6: The percentages of the angles in the four phases of the simulation.

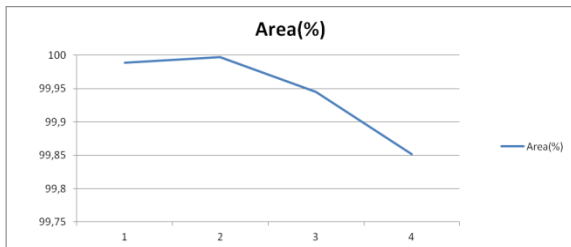


Figure 7: The percentages of the area during the four phases of the simulation.

#### 4.4 Wear of the surfaces

In these simulations the input data are again two point clouds. The first point cloud is not supposed to have suffered any damage, while the second had damages introduced in all three axes. Dimensions, surfaces elements and results of the comparisons are presented in the following tables:

| <i>1<sup>st</sup> pair of surfaces</i> |     | <i>2<sup>nd</sup> pair of surfaces</i> |     |
|--|-----|--|-----|
| <b>Dx (mm)</b>                         | 100 | <b>Dx (mm)</b>                         | 100 |
| <b>Dy (mm)</b>                         | 100 | <b>Dy (mm)</b>                         | 100 |
| <b>Dz (mm)</b>                         | 59  | <b>Dz (mm)</b>                         | 51  |

Table 5. Dimensions of the simulation surfaces

| <i>1<sup>st</sup> pair of surfaces</i>        |     | <i>2<sup>nd</sup> pair of surfaces</i>        |      |
|---|-----|---|------|
| <b>Triangles</b>                              | 39  | <b>Triangles</b>                              | 46   |
| <b>Average triangle area (mm<sup>2</sup>)</b> | 9.5 | <b>Average triangle area (mm<sup>2</sup>)</b> | 14.6 |
| <b>Average triangle side (mm)</b>             | 3.3 | <b>Average triangle side (mm)</b>             | 4.1  |

Table 6. Characteristics of the simulation surfaces

| <b>Points</b>       | 70       |           |
|---------------------|----------|-----------|
| <b>Dx-Dy-Dz(mm)</b> | 0        | up to 0.1 |
| <b>Area (%)</b>     | 100.0000 | 99.9903   |
| <b>Angles (%)</b>   | 100.0000 | 82.5243   |

Table 7. Results from the first pair of compared surfaces

| <b>Points</b>       | 70       |         |
|---------------------|----------|---------|
| <b>Dx-Dy-Dz(mm)</b> | 0        | up to 1 |
| <b>Area (%)</b>     | 100.0000 | 99.9526 |
| <b>Angles (%)</b>   | 100.0000 | 40.7767 |

Table 8. Results from the second pair of compared surfaces

Both results of these simulations can be characterized as successful. In the second simulation several different triangles have been formed. The direction of triangles and vectors has changed and therefore the percentage of angles decreased so much.

#### 4.5 Diversity of shapes

A shattered object generates completely random surfaces. Limits of the algorithm attributed to the shape diversity are detected by this simulation. Four applications of the algorithm were performed, each one with a completely different shape. Once again the input data were two point clouds; the first one with no distortions and the second one with distortions up to 0.1mm along the direction of all three axes. Results of the comparisons are presented in table 9:

|   |       |       |       |       |
|---|-------|-------|-------|-------|
| <b>Dx (mm)</b>                                | 350   | 350   | 350   | 350   |
| <b>Dy (mm)</b>                                | 350   | 350   | 350   | 579   |
| <b>Dz (mm)</b>                                | 193   | 20    | 1100  | 150   |
| <b>Points</b>                                 | 100   | 100   | 100   | 1956  |
| <b>Triangles</b>                              | 160   | 167   | 146   | 3163  |
| <b>Average triangle area (mm<sup>2</sup>)</b> | 816   | 734   | 1093  | 3.9   |
| <b>Average triangle side (mm)</b>             | 31    | 29    | 35.5  | 2.1   |
| <b>Area (%)</b>                               | 99.99 | 99.97 | 99.96 | 99.94 |
| <b>Angles (%)</b>                             | 81.87 | 77.24 | 82.87 | 81.59 |

Table 9. Results of the four applications of the algorithm

In these four applications the percentage of angles is constant and close to 80%, while the percentage of the area is also stable and close to 100%. The algorithm does not depend on the shape of the surface, as the results in different cases are considered similar.

## 5. PRACTICAL APPLICATION

In this section, results of a practical implementation of the algorithm are presented. The fragments used for these tests come from the Laboratory of Geotechnics and Rock Mechanics and are actually cylindrical pieces from test drillings through various rock layers. These specimens have then been tested for shear in the laboratory until they fracture. It is these fractured, eroded and shattered surfaces that were put to test by the algorithm.

Scans were performed with the "Next Engine" 3D laser scanner. Three fragments were scanned: a pair of conjugate fragments 1 & 2 (Figure 10) and one single broken piece, i.e. fragment 3 (Figure 11). A point cloud of the "conjugate" piece of fragment 3 (fragment 4) was created using Matlab.

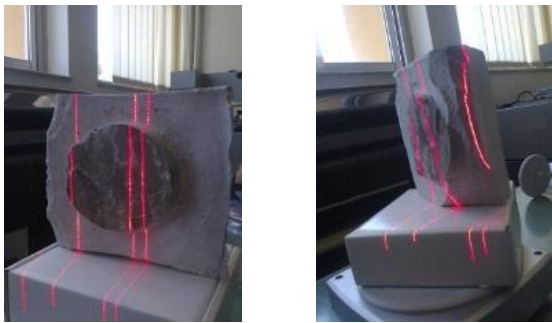


Figure 10. Conjugate fragments 1 & 2 during scanning.

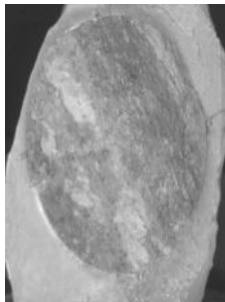


Figure 11. Fragment 3

Two tests were performed. For the first implementation of the algorithm, each cloud consisted of approximately 21000 points. At this high point density, small triangles with an average area of 0.14 mm<sup>2</sup>, were created. Consequently the tolerance angle ( $\varphi$ ) is set at 50 grad. The comparison of both area and angles for fragments 3 and 4 should derive results that reach 100%, since they are mathematically created as conjugate. The next higher results of compatibility are expected from the comparison of fragments 1 and 2, as they are physically conjugate fragments. The results of the test are presented below:

| <i>Fragment</i> | <i>Fragment</i> | <i>Area(%)</i> | <i>Angles(%)</i> |
|-----------------|-----------------|----------------|------------------|
| 1               | 2               | 91.964         | 91.749           |
| 1               | 3               | 93.411         | 92.861           |
| 1               | 4               | 93.411         | 92.861           |
| 2               | 3               | 98.449         | 98.122           |
| 2               | 4               | 98.449         | 98.122           |
| 3               | 4               | 100            | 100              |

Table 10. Results from the first test (conjugate pairs are in the shades lines)

The results of Table 10 are considered successful only for simulated conjugate fragments 3 and 4. On the contrary in cases of physical fragments (1 & 2), the number of triangles created for the comparisons, proves to be very high for the proper operation of the algorithm, as concluded during the simulations. For the second test, the scanned points were decreased to approximately 6000 using Geomagic<sup>®</sup> software. Despite this decimation the areas of interest maintained the essential level of detail. The average area of triangles increased to 0.5 mm<sup>2</sup>. Moreover angle  $\varphi$  tolerance could be reduced to 13 grad. Results from test 2 are being presented in Table 11:

| <i>Fragment</i> | <i>Fragment</i> | <i>Area(%)</i> | <i>Angles(%)</i> |
|-----------------|-----------------|----------------|------------------|
| 1               | 2               | 99.706         | 58.873           |
| 1               | 3               | 98.258         | 47.120           |
| 1               | 4               | 98.258         | 47.120           |
| 2               | 3               | 97.969         | 47.589           |
| 2               | 4               | 97.969         | 47.589           |
| 3               | 4               | 100            | 100              |

Table 11. Results from the second test (conjugated pairs are in the shaded lines)

From the successful results in Table 11 of the practical application using really dubious conjugate surfaces prove that even under adverse conditions the algorithm performs well and that the only critical parameter is the number of 3D points used to describe the surface and form the triangles.

## 6. CONCLUDING REMARKS

### 6.1 Advantages and disadvantages

Summarizing the most important advantages and disadvantages of *Fragmatch* algorithm are being presented. First of all, the algorithm performs a linear comparison between the elements that ensures fast comparisons. Using *Fragmatch* algorithm it is not necessary to use any another software to create triangles and surfaces. The strong point of the algorithm is the use of the normal vectors whose norm is the actual area of the triangle. The algorithm performs fast surface comparison regardless of shape or form. The necessary input data are easy to acquire and are considered standard nowadays, hence the required time for that is very limited. Last but not least, not only global, but also partial matching is being performed.

On the other hand, *Fragmatch* algorithm identifies correspondences between conjugate fragments, but it does not unite the broken pieces. Another disadvantage of the software is that receives as input data only scanned fractured surfaces and not the entire scanned piece. Algorithm's sensitivity is in the number of scan points; the user should be very careful in choosing the right scan density. However, this density may always be decreased later. Finally the size of the input data is limited due to the limitations of the software Matlab.

### 6.2 Proposals

*Fragmatch* algorithm has been created having as primary goals, the linear comparison and the rapid implementation. The algorithm can be extended based on the following proposals.

Determine a correlation coefficient to deal with the multitude of points with the angle section. That factor could eliminate the sensitivity at many scan points. An extension could be made so that broken surfaces could be found automatically by the software. In addition another goal is the unification of pieces and the update of the entire object. Furthermore the transcription of the algorithm in a high level programming language would eliminate the problem with limited size input data. Matching of the pieces could be made more easily by the identification of characteristic forms or by exploiting information about the shape of the final product. Finally an automatic virtual completion of missing pieces could also be added.

## 7. REFERENCES

### References from Journals

1. Brown, B. J. , Toler-Franklin, C., Nehab, D., Burns, M., Dobkin, D., Vlachopoulos, A., Doumas, C., Rusinkiewicz, S. , Weyrich, T., 2010. "A system for high-volume acquisition and matching of fresco fragments: Reassembling Theran wall paintings" *ACM Transactions on Graphics (Proc SIGGRAPH)*, 27(3):84.
2. Da Gama Leitao, H. C. , Stolfi, J., 2002. "A multiscale method for the reassembly of two-dimensional fragmented objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 1239–1251.
3. Freeman, H., Garder, L., 1964. "A pictorial jigsaw puzzles: The computer solution of a problem in pattern recognition," *Electronic Computers, IEEE Transactions on*, vol. EC-13, no. 2, pp. 118–127.
4. Goldberg D., Malon C., Bern M., 2004. "A global approach to automatic solution of jigsaw puzzles" *Comput. Geom. Theory Appl.*, vol. 28, pp.165–174.
5. Huang, Q.-X. , Flory, S., Gelfand, N., Hofer, M. , Pottmann, H.,2006. "Reassembling fractured objects by geometric matching" *ACM Trans. Graphics*, vol. 25, no. 3, pp. 569–578.
6. Karasik, A., Smilansky, U., 2008. "3d scanning technology as a standard archaeological tool for pottery analysis: practice and theory" *Journal of Archaeological Science*, vol. 35, no. 5, pp. 1148–1168.
7. Papaioannou, G. , Karabassi, E.-A., 2003. "On the automatic assemblage of arbitrary broken solid artefacts," *Image and Vision Computing*, vol. 21, no. 5, pp. 401–412.
8. Papaioannou, G., Karabassi, E.-A., Theoharis, T., 2002, "Reconstruction of Three-Dimensional Objects through Matching of Their Parts" *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 24, no. 1, pp.114-124.
9. Papaioannou, G., Karabassi, E.-A., Theoharis, T.,2001. "Virtual archaeologist: Assembling the past" *IEEE Computer Graphics and Applications*, vol. 21, pp. 53–59.
10. Papaodysseus, C., Panagopoulos, T., Exarhos, M., Triantafillou, C., Fragoulis, D.,Doumas, C., 2002. "Contour-shape based reconstruction of fragmented, 1600 bc wall paintings" *Signal Processing, IEEE Transactions on*, vol. 50, no. 6, pp. 1277–1288.
11. Ucoluk, G., Toroslu, I. H., 1999. "Automatic reconstruction of broken 3-d surface objects" *Computers and Graphics*, vol. 23, no. 4, pp. 573–582.
12. Willis, A.R., Cooper, D.B., 2008. "Computational reconstruction of ancient artifacts" *IEEE Signal Processing Magazine*, vol. 25, no. 4, pp. 65–83.

### Other Literature

13. Belenguer, C. S., Vidal, E. V., 2012. "Archaeological Fragment Characterization and 3D Reconstruction based on Projective GPU Depth Maps" *Informatica (CLEI)*, 2012 XXXVIII Conferencia Latinoamericana En, 1-5 Oct. 2012, pp. 1-10, ISBN 978-1-4673-0794-9
14. Gelfand, N., Ikemoto, L., Rusinkiewicz, S., Levoy, M., 2003. "Geometrically Stable Sampling for the ICP Algorithm", Fourth International Conference on 3D Digital Imaging and Modeling (3DIM), pp 260 - 267.
15. Kampel , M., Sablatnig, R., 2004. "On 3d mosaicing of rotationally symmetric ceramic fragments" in *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2 - Volume 02*, ser. ICPR '04. Washington, DC, USA: IEEE Computer Society, pp. 265–268.
16. Koller, D., Levoy, M., 2006. *Computer-aided reconstruction and new matches in the Forma Urbis Romae*, Proc. Third Williams Symposium on Classical Architecture Journal of Roman Archaeology suppl. pp. 103–125.