

Document downloaded from:

<http://hdl.handle.net/10251/52271>

This paper must be cited as:

Vendrell Vidal, E.; Sanchez Belenguer, C. (2014). A Discrete Approach for Pairwise Matching of Archaeological Fragments. *Journal on Computing and Cultural Heritage*. 7(3):1-19. doi:10.1145/2597178.



The final publication is available at

<http://dx.doi.org/10.1145/2597178>

Copyright Association for Computing Machinery (ACM)

# A Discrete Approach for Pairwise Matching of Archaeological Fragments

EDUARDO VENDRELL-VIDAL and CARLOS SÁNCHEZ-BELENQUER, Instituto de Automática e Informática Industrial, Universitat Politècnica de València.

---

This paper addresses the problem of automatic reconstruction of ancient artifacts from archaeological fragments. The technique described here focuses on pairwise matching of flat fragments (typically fresco fragments), and it is intended to be the core of a larger system for artifact reconstruction. Global registration techniques are challenging due to the combinatory explosion that happens in the solution space: the goal is to find the best alignment among all possible ones without an initialization. This fact defines the duality between performance and correction that we face in this work. The proposed technique defines a cost function to evaluate the quality of an alignment based on a discrete sampling of the fragments that ensures data alignment. Starting from an exhaustive search strategy, the technique progressively incorporates new features that lead to a hierarchical search strategy. Convergence and correction of the resulting technique are ensured using an optimistic cost function. Internal search calculations are optimized so the only operations performed are additions, subtractions and comparisons over aligned data. All heavy geometric operations are carried out by the GPU on a pre-processing stage that only happens once per fragment.

Categories and Subject Descriptors: G.1.2 [Numerical Analysis]: Approximation—*Approximation of surfaces and contours*; I.2.10 [Artificial Intelligence]: Vision and Scene Understanding—*Shape*; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—*Curve, surface, solid, and object representations*; I.5.1 [Pattern Recognition]: Models—*Geometric*

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Automatic reconstruction, global registration, GPU, hierarchical

## ACM Reference Format:

Eduardo Vendrell-Vidal and Carlos Sánchez-Belenguer. 2013. A Discrete Approach for Pairwise Matching of Archaeological Fragments. *ACM J. Comput. Cult. Herit.* 1, 1, Article 1 (January 1), 17 pages.

DOI: <http://dx.doi.org/10.1145/0000000.0000000>

---

## 1. INTRODUCTION

Reconstruction of ancient artifacts from fragments found at the archaeological sites, is a tedious task that requires many hours of work from the archaeologist and restoration personnel. Historically, this reconstruction process has been manual, occupying a major proportion of the human effort at excavation sites. In fact, since the assembly work is so time-consuming and labor-intensive, the reconstruction

---

This work is supported by the “Programa de Ayudas de Investigación y Desarrollo (PAID)” of the Universitat Politècnica de València and the “Plan Nacional de I+D+i 2008-2011” from the Ministerio de Economía y Competitividad of Spain, Project ID: HAR2012-38391-C02-02.

Author’s address: E. Vendrell-Vidal, Avenida de los Naranjos, s/n, 46022 Valencia (Spain); email: [even@upv.es](mailto:even@upv.es); C. Sánchez-Belenguer, Avenida de los Naranjos, s/n, 46022 Valencia (Spain); email: [carsanb1@doctor.upv.es](mailto:carsanb1@doctor.upv.es)

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org).

© 1 ACM 1556-4673/1/01-ART1 \$15.00

DOI: <http://dx.doi.org/10.1145/0000000.0000000>

is not even attempted at countless sites around the world, leaving vast quantities of material unstudied and stored indefinitely.

Advancements in low-cost, high-volume acquisition devices and computer systems performance have provided a new tool for archaeologist to face the problem of reconstruction. Operating on digital models of the fragments can rapidly and systematically consider many thousands of possible fragment alignments and combinations. The final goal of these techniques is reducing the amount of candidate matches between fragments, and providing an automatic or semi-automatic tool for archaeologist to recompose the original artifact efficiently.

In the last years, the problem of reassembling fractured 3D objects has gained an increasing importance, and several different approaches have been developed. Most of them perform the search based on the geometry of the fragments, since it represents the only constraint that must be strictly satisfied: one fragment can not penetrate another. In this way, the stated problem is closely related to the challenging problems of shape matching and 3D scan alignment in Computer Graphics and Vision, adding the extra non-penetration constraint.

### 1.1 Problem Statement and Technique Overview

The proposed technique takes as input data two 3D digital models of flat archaeological fragments and outputs the rigid transformation matrices that maximize the contact area between their surfaces. The goal of this paper is not solving the global puzzle, but providing a fast/robust technique to perform the necessary intermediate comparisons between pairs.

Results achieved always ensure that, for the calculated alignment, there are no penetrations between fragments and that the solution corresponds to the global minima.

The kind of fragments considered are very common in archaeological sites: flat fragments that can be disposed on a 2D surface, being a typical example fresco fragments. These are assumed to have their upper and lower faces aligned to the  $X - Z$  plane, being the lower face laying on the  $Y = 0$  plane (Fig. 1). This alignment is normally performed manually during the data acquisition stage, but it can also be automatized by segmenting data as explained in [Huang et al. 2006], and applying least squares to infer the optimal planes that contain the upper and lower faces.

Given these constraints, the solution space to search in has three degrees of freedom, making this approach suitable to solve the same problems faced in [Brown et al. 2008]. The main difference with this technique is that ours optimizes the total alignment error by minimizing a cost function that considers the entire edge of the fragment, instead of a local patch. This allows us to perform a hierarchical search strategy that increases performance considerably, and that would not yield into an enormous speedup in Brown's.

Unlike most of the common approaches in automatic reconstruction, like [Koller and Levoy 2006; Huang et al. 2006], the proposed technique does not perform a forward search using descriptors extracted from the geometry of the fragments. Instead, we discretize the solution space into an equi-spaced set of alignments and we evaluate the quality of every possible combination. The best match between the two compared fragments is the alignment that produces the highest score.

The quality of an alignment is evaluated by using the LCP (*Largest Common Pointset*) metric, which is widely extended in matching techniques and, according to [van Kaick et al. 2010], can be defined as:

$$LCP(P, Q) = \sum_{p \in P} Match(p, Q)$$

$$Match(p, Q) = \begin{cases} 1, & \exists q \in Q, \|p - q\| < \epsilon \\ 0, & \text{otherwise} \end{cases}$$

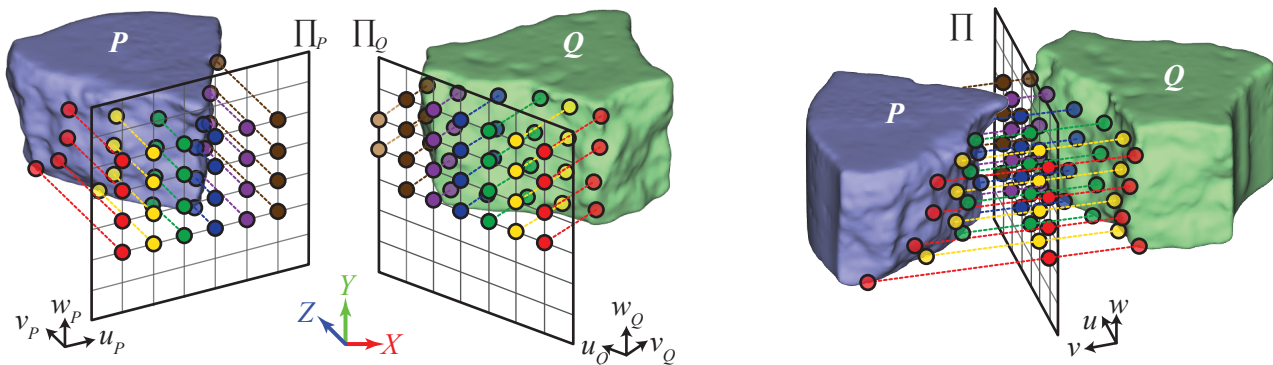


Fig. 1. General overview of the proposed technique: (left) two fragments  $P$  and  $Q$ , with their associated projection planes,  $\Pi_P$  and  $\Pi_Q$ , and the cartesian coordinate system  $(u, v, w)$  associated to each one. Dots represent the calculated samples, colored to facilitate the understanding of the illustration. Lines connecting samples represent the distance between the projection plane and the surface of the fragment. (right) distance measurement between fragments in a given alignment. Notice how samples in both surfaces are aligned, and the distance calculation between facing ones is trivial. 3D models are courtesy by Vienna University of Technology.

where  $P$  and  $Q$  are two fragments,  $p$  is a sampled point in  $P$ ,  $q$  is a sampled point in  $Q$ , and  $\epsilon$  is a tolerance given by the user that indicates if two points are close enough to be considered as matching. The value of  $\epsilon$  depends on how eroded the compared fragments are, and has to be empirically estimated using known matches. Normal values are in the range of [0.1%..2%] of the average fragment size.

To efficiently calculate the corresponding point  $q$  that is aligned to a point  $p$  in the other fragment, we sample uniformly both fragments by using parallel projection planes ( $\Pi_Q$  and  $\Pi_P$ ), which are perpendicular to the  $X - Z$  axis. This way, the degree of freedom associated to the rotation over the  $Y$  axis corresponds to the orientation of the projection plane, and the two degrees of freedom associated to the translation are calculated based on the cartesian system  $(u, w)$  defined by the plane. (Fig. 1).

In order to keep samples aligned, the total amount of discrete displacements over axis  $u$  is determined by the sampling resolution of the fragments, whilst the value of  $v$  remains continuous.

Major contributions on this paper are:

- A discrete characterization of fragments and alignments that allows us to perform all heavy geometric operations by the GPU on a pre-processing stage, and ensures data alignment.
- A hierarchical characterization of fragments.
- A hierarchical search strategy with a very high performance.
- An optimistic cost function that ensures convergency and global correction of the hierarchical search strategy, that operates on the hierarchical characterization of the fragments.

## 1.2 Document Organization

Section 2 gives a background on related previous techniques. In section 3 the proposed technique is explained in detail: subsection 3.1, covers the way 3D models are pre-processed to speed-up search. Subsection 3.2 introduces a simple exhaustive search approach. As expected, achieved results are always correct but execution costs are excessive. To speed up results, while keeping correction, subsections 3.3, 3.4 and 3.5 progressively introduce a hierarchical strategy that ensures convergency to the global solution, while considerably reducing execution cost. In Section 4 achieved results with each technique are presented and compared. Finally, in section 5, major conclusions are highlighted and future improvements to the proposed technique are introduced.

## 2. RELATED WORK

Automatic reconstruction from fragments with computer-aided techniques is a challenge that has been studied for many years [Willis and Cooper 2008]. This discipline can be seen as a specific application of *Surface Registration* techniques, which deal with the most general problem of finding a meaningful relation (or mapping) between the elements of a given input set of shapes [van Kaick et al. 2010].

### 2.1 Registration techniques

In general terms, the most common classification of the existing registration techniques differentiates between *local* and *global* strategies. This distinction is related to the way the solution space is explored: *global* techniques perform a complete search looking for the best solution, whilst *local* techniques perform a partial search, being the achieved results strongly dependent on the initialization.

Almost all global algorithms perform the search using descriptors extracted from the considered models, given that the size of the problem faced makes extremely inefficient to consider all the sampled data. In this respect, multiple descriptors have been proposed based on curvature [Gal and Cohen-Or 2006; Li and Guskov 2005] or integral invariants [Huang et al. 2006; Gelfand et al. 2005]. To calculate the descriptor's alignment, there are techniques based in combinatorial optimization [Gelfand et al. 2005; Okatani and Sugimoto 2005], random algorithms RANSAC [Li and Guskov 2005; Shan et al. 2004] or forward search [Gal and Cohen-Or 2006]. The result of these strategies is a coarse alignment between the input shapes that is generally used as initialization for local search algorithms.

The success of local search strategies is strongly conditioned by the results achieved in the global search stage. Given that in this second stage the goal is to refine the given initialization alignment, the final result may be a local minima. The most prominent example of local search is the ICP (*Iterative Closest Point*) algorithm [Besl and McKay 1992; Chen and Medioni 1992], which alternates between the calculation of correspondence between samples and the calculation of the alignment transformation [Arun et al. 1987]. There are many variations of the original algorithm, that can be found properly classified and detailed in [Rusinkiewicz and Levoy 2001].

### 2.2 Automatic Reconstruction from Fragments

The first approach in this discipline faced the problem of solving jigsaw puzzles [Freeman and Garder 1964]. This kind of problem considers a set of constraints that considerably simplify the correspondence search: each fragment is rectangular, with exactly four neighbors, and pieces fit together via interlocking “indents” and “outdents” [Brown 2008]. In [Wolfson et al. 1988], a solution for bigger puzzles was proposed, also assuming that pieces had four sides, and efficiently calculating the alignment error between each couple of sides. A most recent technique [Goldberg et al. 2004] avoids to assume four-sided pieces, automatically differentiating between “indents”, “outdents” and straight sides.

Contour matching techniques provide solutions for more general problems, without distinguishing between specific edges of features. Applied to 2D domains, some of the most outstanding techniques can be found in [da Gama Leitão and Stolfi 2002; Papaodysseus et al. 2002; Kong and Kimia 2001; Hori et al. 1999] and the references therein. A very common extension of contour matching techniques in archaeological fragment reconstruction are pottery re-assembly techniques. These are applied to revolution surfaces and can take advantage on the additional constraints of axial symmetry, torsion and curvature to guide the search process [Karasik and Smilansky 2008; Willis 2004; Kong and Kimia 2001; Üçoluk and Toroslu 1999].

Surface matching techniques are applied to more general problems, considering 3D input data, and providing solutions to problems with six degrees of freedom. In these kind of techniques the combinatorial explosion in the solution space makes exhaustive searches prohibitive. The first approach in this area was presented in [Papaioannou and Karabassi 2003; Papaioannou et al. 2001], with the under-

lying assumption that the fractured faces were nearly planar and the matched each other completely. Using a projective space, GPU depth maps were analyzed to reconstruct the original object. Then, Sanford's Digital Forma Urbis Romae project [Koller and Levoy 2006] dealt with heavily eroded fragments, whose fractured surfaces sometimes did not even touch each other. In this approach, instead of using the geometry, reconstruction is done by matching incisions on the fragment's top surfaces. The technique described in [Huang et al. 2006] reassembles solid objects by first identifying fractured regions, and then generating clusters of feature patches for alignment-based matching. In addition, this approach also introduces an effective technique for multi-piece global matching of fragments. Finally, [Brown et al. 2008] exploits 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.

As suggested in [Toler-Franklin et al. 2010], search process can be enriched by additional criteria, in addition to the geometric one. Thus, including color and texture information, may provide better results in certain cases [Sagiroglu and Ercil 2006; Fornasiero and Toniolo 2005]. However, when considering multiple properties, a classifier is required to estimate the quality of a given alignment, according to all considered criteria [Funkhouser et al. 2011; Shin et al. 2010].

Multi-piece matching techniques are applied to the global reconstruction of the original object. Taking as input data pairwise matches, these techniques deal with the assembly of larger clusters. The main difficulty is that small errors in alignment between adjacent fragments leads to gaps and interpenetration. To solve this problem, most of the techniques apply a global relaxation step to optimize fragments alignment. Some works on this line can be found in [Castañeda et al. 2011; Zhu et al. 2008; Huang et al. 2006; Goldberg et al. 2004].

The work described in this paper faces the same problem that [Brown et al. 2008] did, but it uses a more complex search strategy: instead of performing an exhaustive analysis, we add some optimizations that considerably accelerate the execution time. For each possible alignment, its quality is evaluated using GPU data, similarly to [Papaioannou and Karabassi 2003; Papaioannou et al. 2001].

### 3. METHOD

**This section covers the details of the proposed technique, which is divided in two stages: pre-processing of the fragments, and searching between pairs.**

**Pre-processing stage is explained in subsection 3.1. Subsection 3.2 introduces a basic exhaustive strategy whose main disadvantage is performance. To speedup the search process, a hierarchical strategy is introduced in subsections 3.3, 3.4 and 3.5, which face the main bottlenecks of the exhaustive approach: orientations, displacements and cost function evaluation, respectively.**

**To prevent correction loss, the proposed approach ensures that the searching process always converges to the optimal solution without stopping in local minima.**

#### 3.1 Fragment Pre-Processing

During the search process, lots of comparisons between pairs of fragments have to be done. These comparisons involve two main operations: (1) uniformly sampling each fragment to calculate distances between its visible surface and the projection plane, and (2) adding these distances between facing samples to compute the quality of each evaluated alignment.

The first type of operation only depends on the topology of each fragment, while the second type depends on which pair of fragments are compared, and the specific alignment evaluated. It is also important to notice that while the first kind of operation requires hard computation, including geometric transformations, visibility tests and discretization operations, the second type of operation can be performed using simple additions. The goal of the pre-processing stage is to perform all these costly

operations that only depend on each fragment, producing results that can be used in as many comparisons as needed in further searches. This way, the search process can be performed by using only simple operations over pre-processed aligned data.

To pre-calculate the projective distances of fragments, the proposed technique uses GPU (*Graphic Processor Unit*) computing capabilities, which is very similar to the approach presented in [Papaioanou et al. 2001]. The main difference with this technique is that we do not make the assumption that the fractured faces are nearly planar and they match each other completely. This way, a major number of cases have to be considered, and makes the approach suitable for more general problems.

GPUs can efficiently solve geometric transformations by using a specific wired unit in the graphics pipeline (Transform Clipping and Lightning). The use of depth buffers, combined with stencil buffers, allows us to perform visibility tests in screen-space with hardware acceleration and, given the discrete nature of generated images, the rasterization unit performs all the heavy calculations needed to get uniform samples over the surface of fragments.

Given these advantages, the way to measure distances to the surface of the fragment consists in calculating the orthographic projection matrix defined by the angle of the plane, and rendering the depth buffer into a frame buffer object. The resolution of the depth buffer used determines the number of samples calculated, and the precision of measures is considerably high.

### 3.2 Exhaustive approach

Exhaustive approach, also known as *naive search*, is the most basic approach. Its main advantage is that it always produces the correct results, avoiding local minima. On the other hand, its main disadvantage is related to performance: it has to evaluate all possible alignments between fragments to find the best match.

To simplify notation, we define an *Alignment*,  $A$ , and its associated cost function,  $c(A)$  as:

$$A(\theta_P, \theta_Q, \delta_u) \tag{1}$$

$$c : \mathbb{R} \times \mathbb{R} \times \mathbb{Z} \rightarrow \{x \in \mathbb{N} : x > 0\} \tag{2}$$

where  $\theta_P$  and  $\theta_Q$  in (1) correspond to the orientation of both fragments, and  $\delta_u$  corresponds to the relative displacement along the  $u$  axis, defined by the projection plane, and illustrated in Fig. 1. The cost function detailed in (2) takes as input data an alignment, and returns a natural number greater than 0 that corresponds to the amount of matching samples in both fragments that are close enough to be considered as matching.

Given that all possible alignments have to be compared, the total amount of evaluations of  $c(A)$  can be expressed as:

$$\mathcal{O}(m, n) \approx m^2 * (2n - 1) \tag{3}$$

where  $m$  is the number of orientations considered, and  $n$  is the number of samples calculated.

The  $2n - 1$  term in (3) represents the total amount of displacements that can be performed, in order to keep samples from both fragments aligned. The  $m^2$  term in (3) means that, for every orientation in fragment  $Q$ , every orientation in fragment  $P$  has to be compared. However, notice how there are lots of pairs of orientations which produce the same results ( $\theta_P = \theta_Q + b$ ).

To reduce the number of combinations calculated, it can be assumed that one fragment ( $P$ ), is only studied for  $k$  equi-spaced orientations, whilst the second ( $Q$ ) is studied for the whole  $m$  orientations, being  $k < m$ . The value of  $k$  has to be great enough to fully characterize  $P$ , but low enough to simplify the search process. **This way,  $k$  is not related to the size of the fragment, but to the convexity of its surface: the more self-occlusions on the surface of the fragment, the bigger value of  $k$ . In practice, values for  $k$  in the range [6..16] have provided results equivalent to  $k = m$ .**

The final amount of alignments to be evaluated can, then, be expressed as:

$$\mathcal{O}(m, n, k) \approx (m * k) * (2n - 1) \quad (4)$$

To speedup the cost function evaluation, each fragment  $P$  is characterized as a set of distance matrices  $P_{\theta_k} = [p_{i,j}]_{|u| \times |w|}$ , where  $\theta_k$  is the orientation of the projection plane,  $|u|$  and  $|w|$  represent the sampling resolution over axes  $U$  and  $W$ , respectively, and each element  $p_{i,j}$  stores the calculated distance between the  $(i, j)^{th}$  sample and the projection plane  $\Pi_{\theta_k}$ . Orientation values of  $\theta_k$  are defined as:  $\theta_k = k * 2\pi/m$ , being  $m$  the discretization resolution on rotations.

Since alignment between samples is ensured by the proposed characterization, given two fragments ( $P, Q$ ) and a discrete displacement of fragment  $Q$  over  $U$  axis ( $\delta_u$ ), the sample facing  $p_{i,j}$  is  $q_{i-\delta_u,j}$ .

$$c(A) = \sum_{i=0}^{|u|-1} \sum_{j=0}^{|w|-1} Match(p_{i,j}, q_{i-\delta_u,j}) \quad (5)$$

$$\delta_v = Min(p_{i,j} + q_{i-\delta_u,j}) \quad (6)$$

$$Match(p_{i,j}, q_{k,j}) = \begin{cases} 1, & p_{i,j} + q_{k,j} - \delta_v \leq \epsilon \\ 0, & \text{Otherwise} \end{cases} \quad (7)$$

To calculate the cost function of an alignment (5), first we need to calculate the maximum displacement  $\delta_v$  of fragment  $Q$  over  $V$  axis that produces contact between the surfaces of fragments (6). Then, given the calculated  $\delta_v$ , for each pair of aligned samples  $p_{i,j} \in P$  y  $q_{i-\delta_u,j} \in Q$ , it has to be evaluated if they are close enough to be considered as matching (7), where  $\epsilon$  is the tolerance introduced by the user.

Notice how the distance between the projection plane and the surface of the fragment does not affect the technique, given that  $\delta_v$  ensures that contact between both fragments always happen.

Figure 2 illustrates the calculation of  $\delta_v$  and  $c(A)$ .

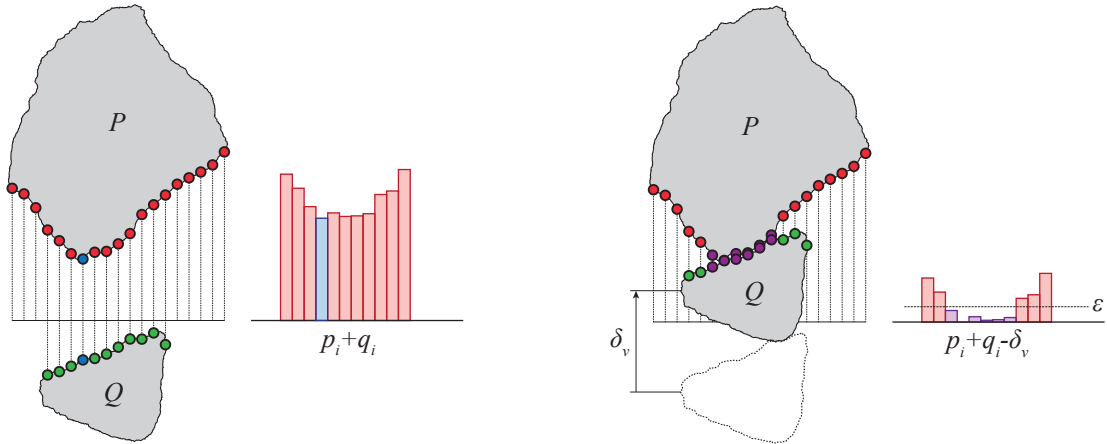


Fig. 2. Cost function evaluation for two centered fragments ( $\delta_u = 0$ ). (left)  $\delta_v$  calculation stage. Minimum distance is marked as blue samples. (right) Score calculation applying the  $\delta_v$  displacement. Matching samples, according to  $\epsilon$ , are marked in purple.

Algorithm 1 shows the pseudo-code to perform an exhaustive search.



**ALGORITHM 1:** Exhaustive search

---

```

best ← null;
for  $\theta_P \leftarrow 0$  to  $2\pi$  step  $2\pi/k$  do
  for  $\theta_Q \leftarrow 0$  to  $2\pi$  step  $2\pi/m$  do
    for  $\delta_u \leftarrow -|u| + 1$  to  $|u| - 1$  step 1 do
       $A \leftarrow \text{Alignment}(\theta_P, \theta_Q, \delta_u)$ ;
      if  $c(A) > c(\text{best})$  then
         $\text{best} \leftarrow A$ ;
return best

```

---

### 3.3 Hierarchical Orientations

The goal of this optimization is to reduce the computational cost associated to the  $(m * k)$  term shown in (4), that is caused by the exhaustive exploration of orientations in the solution space. By introducing a binary hierarchical search, we intend to achieve an execution cost of  $\mathcal{O}(m) \approx \log_2(m)$  for the best case execution scenario.

We propose a hierarchical fragment characterization based on *Levels Of Detail* (LOD), defined as:

- (1) Each fragment is characterized as a set of  $\log_2(m) + 1$  LODs.
- (2) Each  $LOD_x$  covers the full  $2\pi$  orientations, and is made up of  $2^x$  distance matrices.
- (3) Each distance matrix  $P_{[\theta_1..2\pi[} = [p_{i,j}]_{|u| \times |w|}$ , of a given  $LOD_x$  covers  $2\pi/2^x$  orientations.
- (4) Each element of the distance matrix  $p_{i,j}$  stores two values: the maximum and minimum projective distance of the  $(i, j)^{th}$  sample in the covered angular range, which we notate as  $\lceil p_{i,j} \rceil$  and  $\lfloor p_{i,j} \rfloor$ , respectively.

This way, it can be said that  $LOD_1$  refines  $LOD_0$  with two distance matrices  $P_{[0..2\pi[}$  and  $P_{[\pi..2\pi[}$ , which are called *children* of  $P_{[0..2\pi[} \in LOD_0$ , and provide a complete and disjoint partition of their *father*.

The proposed hierarchical search starts by comparing coarse representations of fragments and, recursively, refines the most promising ones until the finest *Level Of Detail* (LOD) is reached. When this happens, the search stops returning the resulting alignment as the result. For the technique to converge to the global solution, avoiding local minima, the cost function of each alignment in an intermediate LOD has to be evaluated in an optimistic way. This makes necessary to extend equations (1) (2) (6) (7) as follows:

$$A\left(\theta_P, \left[\theta_Q^{(a)} \dots \theta_Q^{(b)}\right], \delta_u\right) \quad (8)$$

$$c: \mathbb{R} \times [\mathbb{R} \dots \mathbb{R}] \times \mathbb{Z} \rightarrow \{x \in \mathbb{N} : x > 0\} \quad (9)$$

$$\delta_v = \text{Min}(\lceil p_{i,j} \rceil + \lceil q_{i-\delta_u,j} \rceil) \quad (10)$$

$$\text{Match}(p_{i,j}, q_{k,j}) = \begin{cases} 1, & \lfloor p_{i,j} \rfloor + \lfloor q_{k,j} \rfloor - \delta_v \leq \epsilon \\ 0, & \text{Otherwise} \end{cases} \quad (11)$$

From (8) it can be appreciated that now, we compare a unique orientation of fragment  $P$  with a range of orientations of fragment  $Q$ , which correspond to a distance matrix of a given LOD of  $Q$ . The cost function (9), as it did before, evaluates how many samples are close enough to be considered as matching. To do so, first it calculates  $\delta_v$  using maximum projection distances (10) and then, it evaluates the distance between facing samples using minimum projection distances (11).

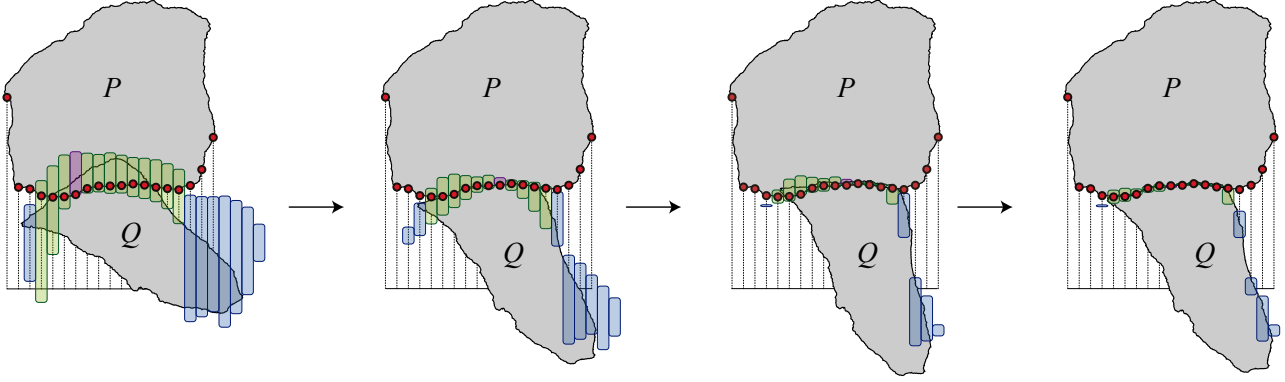


Fig. 3. Angular search process. Fragment  $Q$  is represented in the middle orientation of each distance matrix rotational interval. Purple boxes show the most restrictive distances in each case (which determines the value of  $\delta_v$ ), green boxes show matching samples, and blue boxes show non matching samples. Notice how, in four iterations, ambiguity is highly reduced and the solution converges to the optimal one. Also notice how the value of  $\delta_v$  decreases in each iteration, and how some previously matching samples fail the  $Match()$  test in later alignments.

As it can be appreciated in Fig. 3, samples in *children* distance matrices always have minimum distances greater or equal than their *father*, and maximum distances smaller or equal than their *father*. This means that the value of  $\delta_v$  decreases as the representation is refined, since it uses maximum distances. In the other hand, since the value of the  $Match()$  function depends on  $\delta_v$  and the minimum distances, as the representation is refined the value returned by this function is monotonically decreasing. According to this property of the metric, it can be ensured that the search process converges to the optimal solution without stopping in local minima.

The search process starts by initializing a heap with  $k$  alignments: one for each  $k$  orientation of fragment  $P$ , facing fragment  $Q$  represented in  $LOD_0$ . The heap stores all evaluated alignments sorted in descending order according to their cost function. Each iteration, the alignment ( $A$ ) in the root of the heap (the one with the highest cost function value) is popped and refined into another two alignments in the next LOD as follows:

$$A\left(\theta_P, \left[\theta_Q^{(a)} \dots \theta_Q^{(b)}\right], \delta_u\right) \rightarrow \begin{cases} A_1\left(\theta_P, \left[\theta_Q^{(a)} \dots \theta_Q^{((b-a)/2)}\right], \delta_u\right) \\ A_2\left(\theta_P, \left[\theta_Q^{((b-a)/2)} \dots \theta_Q^{(b)}\right], \delta_u\right) \end{cases} \quad A \in LOD_x : A_1, A_2 \in LOD_{x+1}$$

For each one of the two new alignments the cost function has to be evaluated in all possible displacements. The maximum score achieved is the one we use to insert each new alignment into the heap. This process continues until the root of the heap is an alignment in the finest LOD. In this case, the alignment is returned as the result of the problem. Fig. 3 shows a graphical representation of the proposed search process.

### 3.4 Hierarchical Displacements

In this case, the goal is to reduce the computational cost associated to the  $(2n - 1)$  term shown in (4), caused by the exhaustive exploration of displacements in the solution space, given the orientation of the fragments. By introducing a binary hierarchical search, we intend to achieve an execution cost of  $\mathcal{O}(n) \approx \log_2(n)$  for the best case execution scenario.

As happened with orientations, LODs are used to characterize fragments (Fig. 4 left) as follows:

- (1) Each distance matrix is represented as a set of  $\log_2(n)+1$  displacement LODs, being  $n = \text{Max}(|u|, |w|)$ .
- (2) Each  $LOD_x$  is a distance matrix of size  $2^x \times \left\lceil \frac{2^x |w|}{|u|} \right\rceil$ , if  $|u| \geq |w|$ , or  $\left\lceil \frac{2^x |u|}{|w|} \right\rceil \times 2^x$  if  $|w| > |u|$ .
- (3) Each element  $p(i, j)$  of the distance matrix in  $LOD_x$  stores three values: the maximum projective distance  $\lceil p(i, j) \rceil$  of samples  $\lceil p(k, l) \rceil \in LOD_{x+1}$ , being  $i * 2 \leq k \leq i * 2 + 1$  and  $j * 2 \leq l \leq j * 2 + 1$ , the minimum projective distance  $\lfloor p(i, j) \rfloor$  of samples  $\lfloor p(k, l) \rfloor \in LOD_{x+1}$ , and the total amount of samples  $|p(i, j)|$  contained in  $|p(k, l)| \in LOD_{x+1}$ .

Searching strategy for displacements is very similar to the one for orientations, and the same restrictions in the cost function have to be applied to ensure convergency: it has to be monotonically decreasing as LOD increases. To do so, equations (8) (9) (10) (11) have to be extended as follows:

$$A \left( \theta_P, \left[ \theta_Q^{(a)} \dots \theta_Q^{(b)} \right], \delta_u, LOD_\delta \right) \quad (12)$$

$$c : \mathbb{R} \times [\mathbb{R} \dots \mathbb{R}] \times \mathbb{Z} \times \mathbb{Z} \rightarrow \{x \in \mathbb{N} : x > 0\} \quad (13)$$

$$\delta_v = \text{Min}(\lceil p_{i,j} \rceil + \lfloor q_{i-\delta_u, j} \rfloor) \quad (14)$$

$$\text{Match}(p_{i,j}, q_{k,j}) = \begin{cases} |p_{i,j}|, & |p_{i,j}| + |q_{k,j}| - \delta_v \leq \epsilon \\ 0, & \text{Otherwise} \end{cases} \quad (15)$$

In this case, notice how an alignment (12) now indicates the displacement,  $\delta_u$ , and also the LOD in which this alignment is represented. As happened in (10), in (14) maximum distances are used to calculate  $\delta_v$  whilst in (15), instead of adding single samples for each comparison in the  $\text{Match}()$  function,  $|p_{i,j}|$  is used.

Search process starts with a heap containing only one alignment, in which both fragments are represented in  $LOD_0$ . The heap stores all evaluated alignments sorted in descending order according to their cost function. Each iteration, the alignment ( $A$ ) in the root of the heap is popped and refined into another three alignments in the next displacement LOD as follows:

$$A \left( \theta_P, \left[ \theta_Q^{(a)} \dots \theta_Q^{(b)} \right], \delta_u, LOD_\delta \right) \rightarrow \begin{cases} A_1 \left( \theta_P, \left[ \theta_Q^{(a)} \dots \theta_Q^{(b)} \right], \delta_u * 2 - 1, LOD_{\delta+1} \right) \\ A_2 \left( \theta_P, \left[ \theta_Q^{(a)} \dots \theta_Q^{(b)} \right], \delta_u * 2 + 0, LOD_{\delta+1} \right) \\ A_3 \left( \theta_P, \left[ \theta_Q^{(a)} \dots \theta_Q^{(b)} \right], \delta_u * 2 + 1, LOD_{\delta+1} \right) \end{cases}$$

This means that, when refining an alignment in a given LOD, three *children* alignments are inserted into the heap (Fig. 4 right): a centered one, a left displaced one and a right displaced one. The heap is then sorted according to the cost function of each alignment, and the process continues until the root element is an alignment in maximum LOD. In this case, this is returned as the result.

In Fig. 4 it can be appreciated how convergency for cases B, C, D, E, F, H, J and K can be demonstrated as we did for hierarchical orientations: all facing samples have already been evaluated in the *parent* alignment, in which minimum distances were smaller, maximum distances were greater, and the amount of contained samples were also greater. Cases G and I present an extra ambiguity given that they have two *parents* and, in none of them, all the facing samples have been compared. However, it can be demonstrated that a *child* alignment with two *parents* will always have a cost function smaller or equal than one of them, which is sufficient to ensure convergency in the proposed approach.

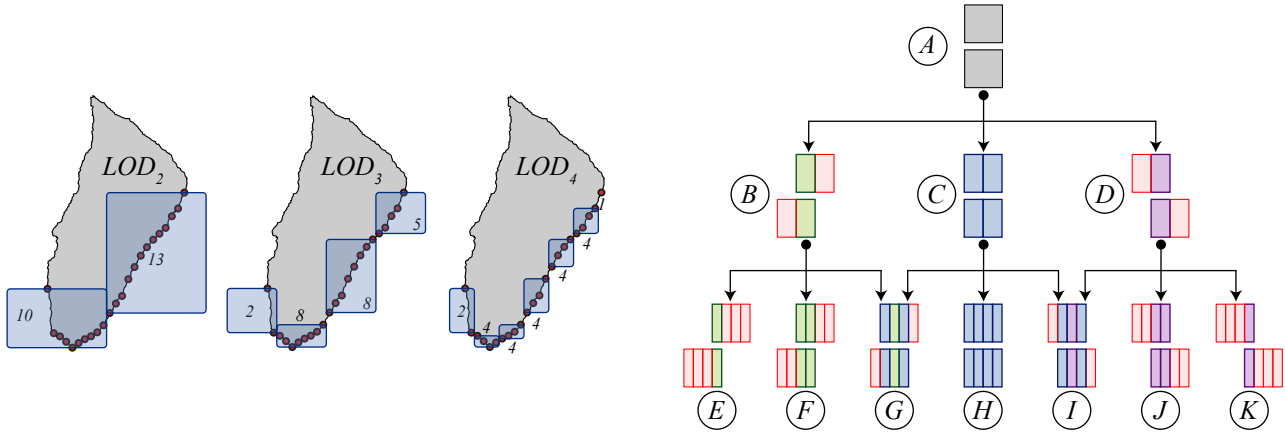


Fig. 4. Hierarchical displacements. (left) A fragment represented in three different LODs. Red dots show samples at maximum resolution, whilst blue boxes represent samples in the illustrated LOD. Associated numbers to each box correspond to the value of  $|p_{i,j}|$ . (right) Search process from  $LOD_0$  to  $LOD_2$ . Each box represents a sample. Marked in green are the samples compared in case *B*, in blue the ones compared in case *C*, and in purple the ones compared in case *D*. Marked in red are the samples without a corresponding one in the other fragment.

### 3.5 Full Hierarchical search

Given the previous two optimizations, it makes sense to combine both to perform a full hierarchical search. The technique proposed here is an orientation-driven search that, in an outer loop hierarchically explores the orientation of both fragments and, for each alignment in a given  $LOD_\theta$ , an inner loop hierarchically explores displacements. The goal of the inner loop is to calculate the value of  $\delta_u$  that maximizes the cost function of the alignment passed by the outer loop, whilst the goal of the outer loop is to find the global solution to the stated problem.

Each iteration of the outer loop (Algorithm 2) pops an alignment from the heap, and inserts two new alignments. Each child alignment is passed to the inner loop (Algorithm 3) in order to compute its maximum value of the cost function and, according to this value, is stored in the sorted heap of active alignments in the outer loop. When the best active alignment is in the maximum orientation LOD, it is returned as the solution to the problem.

By combining these two optimizations, the excessive amount of evaluations of the cost function proposed in the exhaustive approach  $\mathcal{O}(m, n, k) \approx (m*k)*(2n-1)$  gets reduced to  $\mathcal{O}(m, n) \approx \log_2(m)*\log_2(n)$  in the best execution case scenario. However, the third bottleneck associated to the evaluation of the cost function (which has to perform  $|u| * |w|$  calculations per alignment) is still not solved.

To do so, and without loss of correction, when evaluating displacements of an alignment in the inner loop, instead of searching until the maximum LOD we only search until  $LOD_\theta = LOD_\delta$ . This allows to perform very fast evaluations of the cost function when the uncertainty of the representation is high, and progressively achieve more precise results, as we approach to the global solution. Given that the proposed cost function is optimistic, convergency is ensured and, despite that the lack of precision in coarse representations may lead to perform some unnecessary comparisons, the increase of global performance fully justifies the proposed optimization, as we show in the next section.

**ALGORITHM 2:** Hierarchical orientation-driven search

---

```

H ← Heap() :
for  $\theta_P \leftarrow 0$  to  $2\pi$  step  $2\pi/k$  do
  | H.Push(A( $\theta_P$ , [0.. $2\pi$ ], 0, 0));
while true do
  | A( $\theta_P$ , [ $\theta_Q^{(a)} \dots \theta_Q^{(b)}$ ],  $\delta_u$ , 0) ← H.Pop();
  | if  $A.LOD_\theta = LOD_{\theta_{max}}$  then
  |   | return A;
  | else
  |   |  $A_1 \leftarrow Alignment(\theta_P, [\theta_Q^{(a)} \dots \theta_Q^{((b-a)/2)}], \delta_u, 0)$ ;
  |   |  $A_2 \leftarrow Alignment(\theta_P, [\theta_Q^{((b-a)/2)} \dots \theta_Q^{(b)}], \delta_u, 0)$ ;
  |   | H.Push(BestDisplacement(A1));
  |   | H.Push(BestDisplacement(A2));

```

---

**ALGORITHM 3:** Hierarchical displacement search *BestDisplacement(A)***Data:** An alignment *A***Result:** The best displacement for orientations specified in *A*


---

```

begin
  | H ← Heap() :
  | H.Push(A);
  | while true do
  |   | A( $\theta_P$ , [ $\theta_Q^{(a)} \dots \theta_Q^{(b)}$ ],  $\delta_u$ ,  $LOD_\delta$ ) ← H.Pop();
  |   | if  $A.LOD_\delta = A.LOD_\theta$  then
  |   |   | return A;
  |   | else
  |   |   |  $A_1 \leftarrow Alignment(\theta_P, [\theta_Q^{(a)} \dots \theta_Q^{(b)}], \delta_u * 2 - 1, LOD_{\delta+1})$ ; H.Push(A1, c(A1));
  |   |   |  $A_2 \leftarrow Alignment(\theta_P, [\theta_Q^{(a)} \dots \theta_Q^{(b)}], \delta_u * 2 + 0, LOD_{\delta+1})$ ; H.Push(A2, c(A2));
  |   |   |  $A_3 \leftarrow Alignment(\theta_P, [\theta_Q^{(a)} \dots \theta_Q^{(b)}], \delta_u * 2 + 1, LOD_{\delta+1})$ ; H.Push(A3, c(A3));

```

---

## 4. RESULTS

This section empirically evaluates the correction and performance of the proposed technique using a set of ceramic fragments compound by 21 pieces. It is important to notice that comparisons are performed considering only pairs of fragments, and results are expressed as the average search time for each possible combination. The goal is, then, to prove the performance and correction for pairwise comparisons, no matter how big the set of fragments is.

For larger datasets, where the combinatory explosion of potential matches can lead to a great performance loss, a higher level global search strategy should be implemented to efficiently discard potentially bad pairs of fragments. This problem is out of the scope of the proposed technique, but can be implemented easily using the concepts explained here: if the orientation-driven search illustrated in Algorithm 2 is initialized with all the possible pairs of fragments, the first result achieved will be the best pair among all possible ones. Given that each iteration of the hierarchical algorithm the most promising alignment is selected and refined, potentially bad alignments are automatically discarded.

This way, with a different initialization, this technique can be extended to hierarchically evaluate all possible pairs ( $N$  to  $N$  matching instead of 1 to 1).

#### 4.1 Description

To evaluate the efficiency of the proposed technique, a set of flat ceramic fragments from the seventeenth century has been used. As it can be seen in Fig. 7 (left) fragments are very eroded, and the fractured edges do not match between them perfectly.

All test have been executed in a 3.4 GHz Intel Core i7 computer with 4GB of RAM. The 3d models of the fragments have been acquired using a Konica Minolta Vivid 9i laser scanner and a turning plate. The average acquisition time, including partial view registration and global remeshing of the scanned point-clouds, has been around an hour per fragment. Resulting 3d files present an average triangle count of about 100K triangles.

An example of the 3d models used for evaluation purposes is shown in Fig. 5.

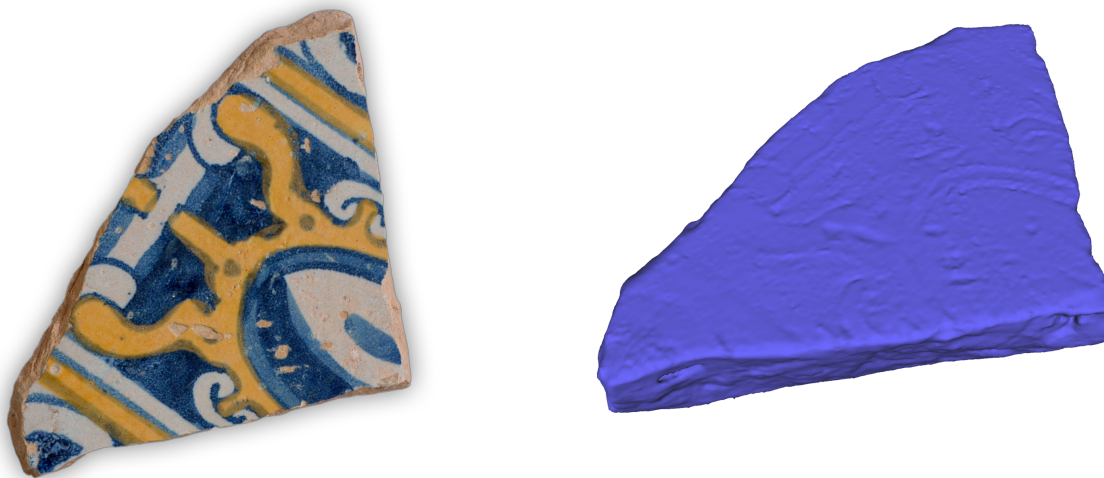


Fig. 5. A real fragment (left) and the acquired 3D model (right).

Each fragment has been pre-processed once per each resolution of the evaluations performed. The maximum pre-processing time has been below 20 seconds, when calculating 1024 orientations with 1024 sample points in the  $U$  axis and 16 points in the  $W$  axis. Considering that, in a normal execution, one fragment has to be pre-processed only once and that the acquisition time is much bigger, these times can be considered as not significant.

In order to compare among the different proposed techniques, 36 different searches between all pairs of fragments have been executed. Each search is characterized by the amount of orientations considered (from  $2^5$  to  $2^{10}$ ) and the sampling resolution on the fragment's surface over axis  $U$  (from  $2^5$  to  $2^{10}$ ). Fig. 6 shows the average execution time for each technique.

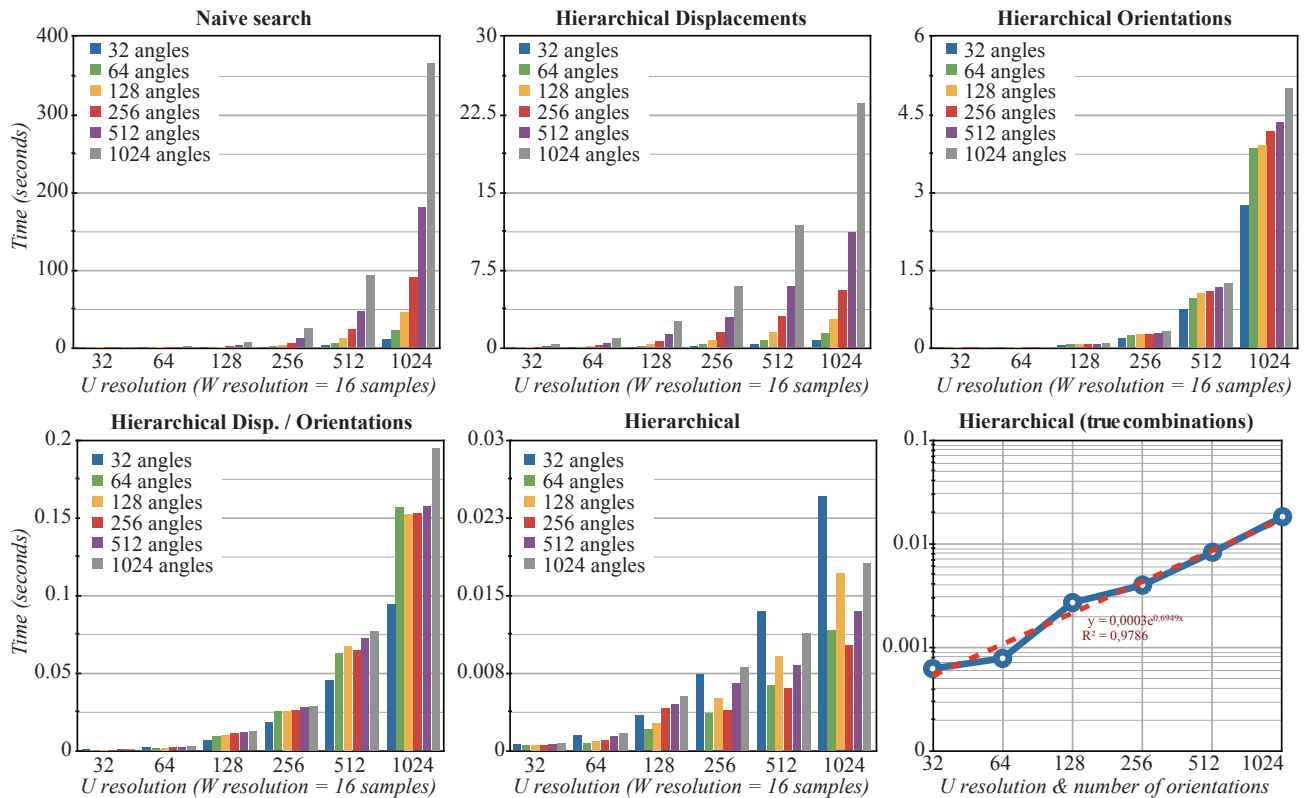


Fig. 6. Comparison between execution times obtained with the proposed techniques for different input data sizes.

## 4.2 Evaluation

Given that the goal of this approach is to solve the pairwise matching of fragments, and not to solve the global puzzle, a result is considered to be correct if the transformation returned is the global minima of the stated cost function. Global correction considering all the fragments is out of the scope of this technique, and is highly conditioned by the ambiguity of the set of fragments considered: a higher level technique to reconstruct the original artifact should discard the matching pairs that contradict the common goal.

During the tests, all techniques have returned exactly the same results for each couple of fragments compared. This was expected given that, in this paper, it has been formally demonstrated that the hierarchical strategy always find the global minima solution.

Local minima results only exist when the sampling resolution is too low, given the irregularities of fragments' surface. In the evaluation set used, the maximum fragment size was smaller than 20 centimeters meaning that, with a sampling resolution over  $U$  axis of 1024, the separation between samples was always smaller than 0,019 millimeters (which is similar to the scanner resolution). When a local minima solution appears, due to under-sampling, it appears in all proposed techniques.

After performing all the tests, the only difference between proposed techniques was the execution time: naive search (top-left) shows an exponential computing cost, that increases regularly as the size of the problem increases. Hierarchical displacements (top-center) attenuates the effect that causes the increase of samples considered, and makes the execution time more linear depending on the resolution

over  $U$  axis (notice that the  $X$  axis of the charts uses an exponential scale). By using hierarchical orientations (top-right), the effect of the number of angles considered is reduced significantly, being the final cost almost logarithmic to the amount of orientations. When these two techniques are combined (bottom-left), both improvements are combined, and the overall execution time behaves almost linearly to the global size of the problem.

Finally, if the cost function evaluation is optimized by stopping the inner displacement search when  $LOD_\theta = LOD_\delta$ , we achieve the final hierarchical technique (bottom-center), which speeds-up the naive results more than 20.000 times in the most complex problem, and provides the technique an execution cost linear to the size of the problem. Given that, in the proposed hierarchical technique, the displacement and orientation searches evolve simultaneously, only problems with the same resolution in both parameters are efficiently evaluated. If we consider only these combinations (bottom-right), it can be seen how the resulting technique behaves linearly to the size of the problem:  $\mathcal{O}(n) \approx n$ , being  $n$  the resolution of angles/displacements considered.

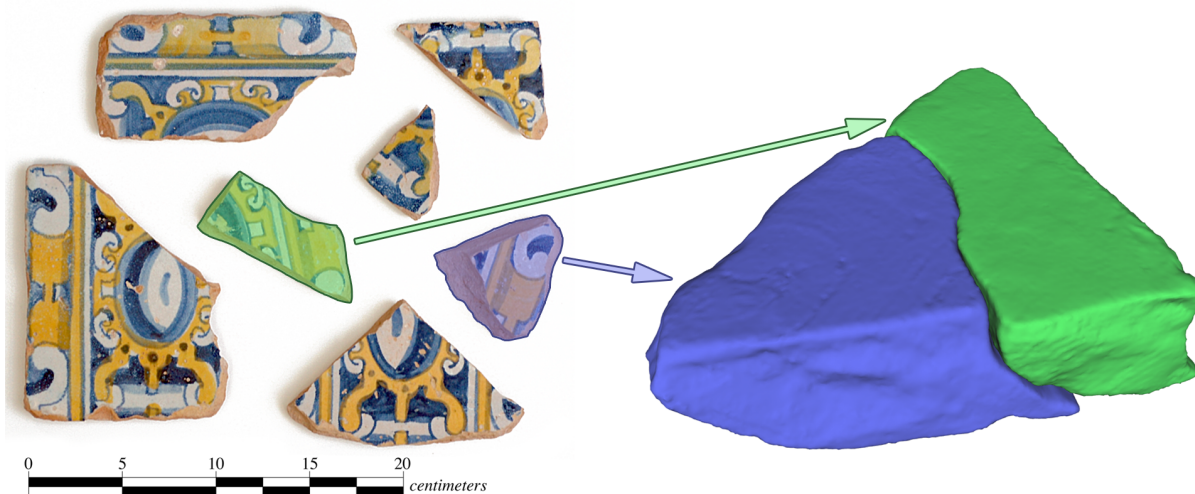


Fig. 7. Evaluation with real fragments. (left) Some of the fragments used to evaluate the proposed technique. (right) One of the results achieved during the search process.

## 5. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a discrete global registration technique applied to the automatic reconstruction of archaeological flat fragments.

Starting from a simple cost function and an exhaustive search strategy, several improvements have been proposed to speed up results without loss of correction. By introducing a hierarchical characterization of fragments, and adapting the cost function and search strategy to this data, a hierarchical technique has been developed that only uses addition, subtraction and comparison operations to perform the search. All heavy calculations are executed by the GPU in the pre-processing stage.

Theoretical and empirical evaluations of the proposed strategy have shown a huge performance increase, that makes the resulting technique behave linearly to the size of the problem.



In future works two main extensions to the proposed technique are going to be done: (1) parallelize the search algorithm to speed-up the execution by running the technique in a GPGPU (General Purpose GPU) and (2) extend the technique to handle full 3D models without rotary restrictions.

First optimization is intended to focus in the cost function calculation, given that each pair of facing samples can be independently computed. Since the search loop is hierarchical, and dependent on previous results, a computing restriction is added that makes parallelization very hard to code, and that affects performance dramatically.

Second optimization implies a combinatory explosion in the solution space that may increase considerably the execution time. However, given that new degrees of freedom impose new restrictions on the possible solutions, ambiguity during the search process is expected to decrease faster than in the current approach. In fact, previous implementations with two dimensional data, presented much higher ambiguity (and more searching iterations) than the results shown in this paper.

#### REFERENCES

- K.S. Arun, T. S. Huang, and S. D. Blostein. 1987. Least-Squares Fitting of Two 3-D Point Sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on PAMI-9*, 5 (1987), 698–700.
- P. J. Besl and N. D. McKay. 1992. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (February 1992), 239–256. Issue 2.
- B. J. Brown. 2008. *Registration and Matching of Large Geometric Datasets for Cultural Heritage Applications*. Ph.D. Dissertation. Princeton University.
- B. J. Brown, C. Toler-Franklin, D. Nehab, M. Burns, D. Dobkin, A. Vlachopoulos, C. Doumas, S. Rusinkiewicz, and T. Weyrich. 2008. A system for high-volume acquisition and matching of fresco fragments: reassembling Thera wall paintings. *ACM Trans. Graph.* 27, 3, Article 84 (Aug. 2008), 9 pages.
- A. G. Castañeda, B. J. Brown, S. Rusinkiewicz, T. Funkhouser, and T. Weyrich. 2011. Global consistency in the automatic assembly of fragmented artefacts. In *Proceedings of the 12th International conference on Virtual Reality, Archaeology and Cultural Heritage (VAST'11)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 73–80.
- Y. Chen and G. Medioni. 1992. Object modelling by registration of multiple range images. *Image Vision Comput.* 10 (April 1992), 145–155. Issue 3.
- H. C. da Gama Leitão and J. Stolfi. 2002. A Multiscale Method for the Reassembly of Two-Dimensional Fragmented Objects. *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (September 2002), 1239–1251. Issue 9.
- M. Fornasier and D. Toniolo. 2005. Fast, robust and efficient 2D pattern recognition for re-assembling fragmented images. *Pattern Recogn.* 38 (November 2005), 2074–2087. Issue 11.
- H. Freeman and L. Garder. 1964. Apictorial Jigsaw Puzzles: The Computer Solution of a Problem in Pattern Recognition. *Electronic Computers, IEEE Transactions on EC-13*, 2 (April 1964), 118–127.
- T. Funkhouser, H. Shin, C. Toler-Franklin, A. G. Castañeda, B. J. Brown, D. Dobkin, S. Rusinkiewicz, and T. Weyrich. 2011. Learning how to match fresco fragments. *J. Comput. Cult. Herit.* 4, 2, Article 7 (Nov. 2011), 7:1–7:13 pages.
- R. Gal and D. Cohen-Or. 2006. Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph.* 25 (January 2006), 130–150. Issue 1.
- N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. 2005. Robust global registration. In *Proceedings of the third Eurographics symposium on Geometry processing (SGP '05)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, Article 197, 197–206 pages.
- D. Goldberg, C. Malon, and M. Bern. 2004. A global approach to automatic solution of jigsaw puzzles. *Comput. Geom. Theory Appl.* 28 (June 2004), 165–174. Issue 2-3.
- K. Hori, M. Imai, and T. Ogasawara. 1999. Joint Detection for Potsherds of Broken Earthenware. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* 2 (1999), 2440–2445.
- Q. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann. 2006. Reassembling Fractured Objects by Geometric Matching. *ACM Trans. Graphics* 25, 3 (2006), 569–578.
- A. Karasik and U. Smilansky. 2008. 3D scanning technology as a standard archaeological tool for pottery analysis: practice and theory. *Journal of Archaeological Science* 35, 5 (2008), 1148–1168.
- D. Koller and M. Levoy. 2006. *Computer-aided reconstruction and new matches in the Forma Urbis Romae*. Vol. Supplement. 103–125.
- ACM Journal on Computing and Cultural Heritage, Vol. 1, No. 1, Article 1, Publication date: January 1.

- W. Kong and B. B. Kimia. 2001. On Solving 2D and 3D Puzzles Using Curve Matching. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on 2* (2001), 583–590.
- X. Li and I. Guskov. 2005. Multi-scale features for approximate alignment of point-based surfaces. In *Proceedings of the third Eurographics symposium on Geometry processing*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, Article 217.
- I. S. Okatani and A. Sugimoto. 2005. Globally convergent range image registration by graph kernel algorithm. In *In International Conference on 3D Digital Imaging and Modeling*. 377–384.
- G. Papaioannou, E. Karabassi, and T. Theoharis. 2001. Virtual Archaeologist: Assembling the Past. *IEEE Computer Graphics and Applications* 21 (2001), 53–59.
- G. Papaioannou and E. A. Karabassi. 2003. On the automatic assemblage of arbitrary broken solid artefacts. *Image and Vision Computing* 21, 5 (2003), 401–412.
- C. Papaodysseus, T. Panagopoulos, M. Exarhos, C. Triantafillou, D. Fragoulis, and C. Doumas. 2002. Contour-shape based reconstruction of fragmented, 1600 BC wall paintings. *Signal Processing, IEEE Transactions on* 50, 6 (June 2002), 1277–1288.
- S. Rusinkiewicz and M. Levoy. 2001. Efficient Variants of the ICP Algorithm. In *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*.
- M. S. Sagioglu and A. Ercil. 2006. A Texture Based Matching Approach for Automated Assembly of Puzzles. *Pattern Recognition, International Conference on 3* (2006), 1036–1041.
- Y. Shan, B. Matei, H. S. Sawhney, R. Kumar, D. Huber, and M. Hebert. 2004. Linear Model Hashing and Batch RANSAC for Rapid and Accurate Object Recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition*. 121–128.
- H. Shin, C. Doumas, T. Funkhouser, S. Rusinkiewicz, K. Steiglitz, A. Vlachopoulos, and T. Weyrich. 2010. Analyzing fracture patterns in theran wall paintings. In *Proceedings of the 11th International conference on Virtual Reality, Archaeology and Cultural Heritage (VAST'10)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 71–78.
- C. Toler-Franklin, B. J. Brown, T. Weyrich, T. Funkhouser, and S. Rusinkiewicz. 2010. Multi-feature matching of fresco fragments. *ACM Trans. Graph.* 29, 6, Article 185 (Dec. 2010), 12 pages.
- O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. 2010. A Survey on Shape Correspondence. In *Proc. of Eurographics State-of-the-art Report*. 1–24.
- A. R. Willis. 2004. *Stochastic 3d geometric models for classification, deformation, and estimation*. Ph.D. Dissertation. Providence, RI, USA. Advisor(s) Cooper, David B. AAI3134376.
- A. R. Willis and D. B. Cooper. 2008. Computational reconstruction of ancient artifacts. *IEEE Signal Processing Magazine* 25, 4 (July 2008), 65–83.
- H. Wolfson, E. Schonberg, A. Kalvin, and Y. Lamdan. 1988. Solving jigsaw puzzles by computer. *Ann. Oper. Res.* 12 (February 1988), 51–64. Issue 1-4.
- L. Zhu, Z. Zhou, and D. Hu. 2008. Globally Consistent Reconstruction of Ripped-Up Documents. *IEEE Trans. Pattern Anal. Mach. Intell.* 30 (January 2008), 1–13. Issue 1.
- G. Üçoluk and I. H. Toroslu. 1999. Automatic reconstruction of broken 3-D surface objects. *Computers and Graphics* 23, 4 (1999), 573–582.

Received May 2012; revised ?; accepted ?

# Online Appendix to: A Discrete Approach for Pairwise Matching of Archaeological Fragments

EDUARDO VENDRELL-VIDAL and CARLOS SÁNCHEZ-BELENGUER, Instituto de Automática e Informática Industrial, Universitat Politècnica de València.

## A. DEMONSTRATION OF CONVERGENCY FOR THE HIERARCHICAL DISPLACEMENTS APPROACH

To ensure convergency in the hierarchical displacements strategy is necessary to prove that a *child* alignment with two *parents* will always have a cost function smaller or equal than one of them. In Fig. 4 it can be appreciated how convergency for cases B, C, D, E, F, H, J and K can be demonstrated as we did for hierarchical orientations. However, cases G and I are more complicated.

Considering that both cases are symmetrical, we focus the demonstration using case G, shown in detail in Fig. 8, where the nomenclature used for the next equations is explained.

The first thing to prove is that the value of  $\delta_v$  decreases as the alignment is refined. For this, we take the facing samples from case C and compare them with the ones from case G as follows:

$$\begin{aligned} ([c_1] \geq [g_1]) \wedge ([c'_1] \geq [g'_1]) &\rightarrow [c_1] + [c'_1] \geq [g_1] + [g'_1] \\ ([c_2] \geq [g_3]) \wedge ([c'_2] \geq [g'_3]) &\rightarrow [c_2] + [c'_2] \geq [g_3] + [g'_3] \end{aligned}$$

From these equations it can be said that:

$$\begin{aligned} \delta_v^{C_1} &\geq \delta_v^{G_1} \\ \delta_v^{C_2} &\geq \delta_v^{G_3} \end{aligned}$$

which leads to the conclusion

$$\begin{aligned} \text{Min}(\delta_v^{C_1}, \delta_v^{C_2}) &\geq \text{Min}(\delta_v^{G_1}, \delta_v^{G_2}, \delta_v^{G_3}) \\ \delta_v^C &\geq \delta_v^G \end{aligned}$$

no matter which value takes  $\delta_v^{G_2}$ .

Once proven that the value of  $\delta_v$  decreases as the alignment is refined, to prove that the cost function value also decreases, we distinguish between two cases: 1)  $|C_1| \neq 0$ , and 2)  $|C_1| = 0$ .

The first one is easier to demonstrate because, if  $C_1$  has matching samples, it can be said that:

$$\begin{aligned} |G_1| + |G_2| &\leq |C_1| \\ |G_3| &\leq |C_2| \end{aligned}$$

For the second one, where  $C_1$  has no matching samples, it can be demonstrated that:

$$\begin{aligned} |G_1| &= 0 \\ (|G_2| + |G_3| \leq |C_2|) \vee (|G_2| + |G_3| \leq |B_1|) \end{aligned}$$

considering that the fragment is a closed 3D model, without discontinuities.

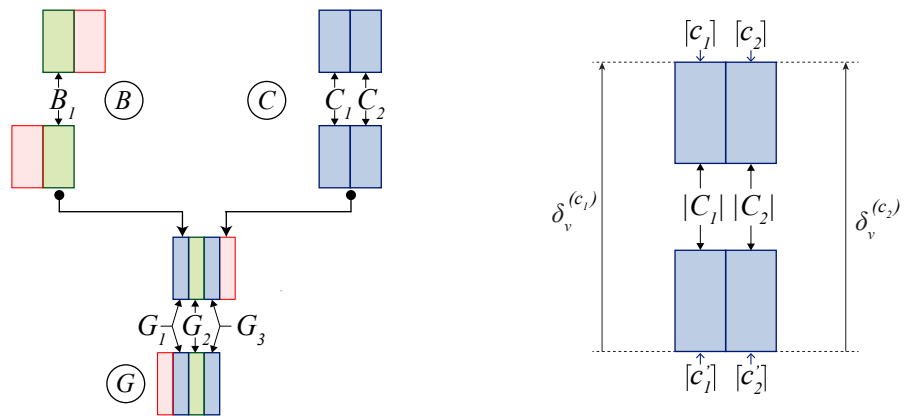


Fig. 8. Demonstration of convergency for case G. (left) both *parents*, B and C of case G, where matching samples colored with the same criteria as Fig. 4. (right) nomenclature used for the demonstration, taking as example case C:  $C_i$  represents two corresponding samples,  $c_i$  and  $c'_i$ ,  $|C_i|$  represents the total number of matching samples.  $\delta_v^{(C_i)}$  represents the distance between the maximum projective distances  $[c_i]$  and  $[c'_i]$ .