

Automatic Reconstruction of Archaeological Finds – A Graphics Approach

Georgios Papaioannou

Department of Informatics, University of Athens
TYPA Buildings, Panepistimiopolis, Ilisia, 15784, Athens, Greece
e-mail: georgep@di.uoa.gr

Evaggelia-Aggeliki Karabassi

Department of Informatics, University of Athens
TYPA Buildings, Panepistimiopolis, Ilisia, 15784, Athens, Greece
e-mail: aggeliki@di.uoa.gr

Theoharis Theoharis

Department of Informatics, University of Athens
TYPA Buildings, Panepistimiopolis, Ilisia, 15784, Athens, Greece
e-mail: theotheo@di.uoa.gr

Abstract Reconstruction of archaeological finds from fragments, is a tedious task requiring many hours of work from the archaeologists and restoration personnel. Up to now, computers have significantly simplified this work by providing tools for the data encoding, storage, classification and visualisation in some cases. In this paper we go one step further by presenting a semi-automatic procedure for the full reconstruction of the original objects using computer graphics and artificial intelligence algorithms on geometrical information.

Keywords: depth buffer, simulated annealing, 3D matching, restoration, global optimisation

1. Introduction

Up to now, computer aided restoration of archaeological sites and historical monuments has focused on four areas: visual representation of historical sites, archiving and visualisation of scanned objects, image processing and restoration of frescos and reconstruction of certain well-structured objects based on feature classification. Examples of such work can be found in [1],[2].

Not much has been done toward the goal of automatic reconstruction of complete objects from their parts. Current algorithms rely heavily on classification of certain qualitative features of the fragments and use description languages to explicitly define the structure of the final, complete object, which must be known a priori, as in [3]. These algorithms are not general and are applicable only to specific types of archaeological fragments that have to be extensively labelled and categorised beforehand.

In this paper we propose a complete method for the full reconstruction of archaeological finds from 3D scanned fragments. The whole procedure involves the automatic detection of candidate fractured faces, the one-by-one fragment matching and the final assembly (clustering) of the fragments into complete or partially complete entities. The reconstruction relies mainly on the geometry of the fragments as provided by polygonal models of the range-scanning of the original fragments. An optional set of constraints, such as material or structural attributes, can significantly improve the overall accuracy and performance. Human intervention is not required but can clarify the final reconstruction result by interactively imposing restrictions on the final combination of objects.

Our technique has been designed to assist archaeologists in the reconstruction of monuments or smaller finds and is integrated in a simulation environment to

avoid unnecessary experimentation with fragile and often heavy fragments. Due to the deteriorated state of most archaeological finds, an automated procedure can not always replace the archaeology expert. However, it provides a useful estimation of valid fragment combinations.

It should be mentioned that the method is not limited to archaeological finds as it does not rely on specific archaeological information and can therefore be applied in other "puzzle-matching" applications regarding solid objects.

2. Method Overview

The reconstruction procedure is divided into three main stages (Fig. 1). At the first stage (preprocessing), each raw polygonal mesh of the 3D-scanned object is partitioned into "facets", i.e. areas of similar face orientation based on the polygon connectivity and the face normal distribution. The bumpiness of each facet is measured using a depth buffer approach discussed in section 3 and the relatively more bumpy facets are marked as candidate test sides for the pairwise matching of the next stage.

In the second stage (off-line reconstruction), the matching error is calculated for all the candidate facets of every possible pair of fragments, by optimising the surface similarity error between two fragments with regard to their relative orientation. Each pair of fragments is examined in order to determine their relative orientation that corresponds to the best fit and therefore the minimal matching error per facet pair. The optimisation is performed using an appropriately adapted version of simulated annealing. The estimation of the complementary surface matching error, which we seek to minimise, can produce high accuracy results and is described in section 4. Not all possible pairs of fragments enter this process, as many are

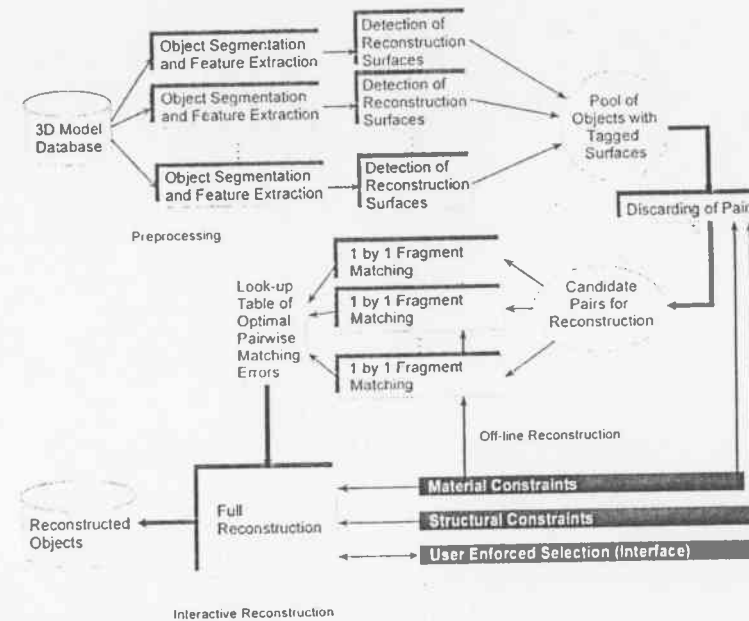


Figure 1. The Fragment reconstruction system architecture.

discarded due to incompatible material, structural or other properties, if such information is available a priori.

When all optimal pairwise (fragment facet by fragment facet) matching error values have been calculated and stored, the third stage (interactive reconstruction), which is a genetic-like algorithm, selects those fragment combinations that minimise an overall error. At this stage, external constraints, such as object material characteristics or a priori structural knowledge, may contribute to the correct fragment clustering.

The next three sections expand on the three parts of the process of reconstruction.

3. Object Segmentation

Prior to defining the best relative matching orientation between two fragments, we should detect the broken facets of each fragment, i.e. the candidate facets for matching.

As a preprocessing stage, every 3D-scanned mesh is partitioned into areas of adjacent nearly coplanar polygons, corresponding to the 'facets' of the object (Fig. 2b). The method used for this segmentation [4] is an iterative region growing technique, which requires no special polygonal mesh representation apart from the raw triangle data. Our triangle-based approach can easily be extended to arbitrary planar polygons.

Once the facets are defined, we proceed by labelling as potential facets for matching those facets that exhibit greater bumpiness (or coarseness), as broken sides tend to be less smooth even if they have been weathered by time. Obviously,

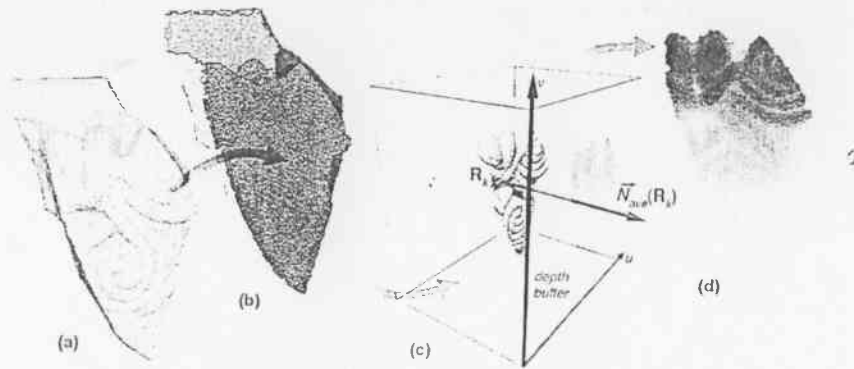


Figure 2. Surface segmentation and coarse faces detection. (a) The original mesh. (b) Partitioned mesh surface. (c) Extraction of the elevation map of a facet. (d) The area of the elevation map over which the facet bumpiness is estimated.

engraved facets are selected as well but these are discarded in the next stage because they produce high matching error.

A facet's bumpiness can not be accurately measured using geometrical information (unless the surface is uniformly sampled) because each facet consists of triangles of varying area. Instead, we use an image-based bumpiness measure calculated on the elevation map of the facet. The elevation map is a two-dimensional array, which represents the distance of the facet from a plane perpendicular to the average facet normal, measured at equidistant grid points (Fig. 2d). This map can easily be obtained from the depth buffer, if the facet triangles are rendered with the viewing direction parallel to the facet normal (Fig. 2c).

We define the bumpiness measure B_{R_i} for a facet R_i as:

$$B_{R_i} = \frac{1}{N_{depth}} \sum_{(u,v) \in d} |\nabla^2 d_{R_i}(u,v)|$$

where $d_{R_i}(u,v)$ is the elevation map of facet R_i and N_{depth} is the number of non-infinite values in $d_{R_i}(u,v)$. B_{R_i} reflects the average steep transitions per surface facet.

$\nabla^2 d_{R_i}(u,v)$ is the common Laplacian image operator.

4. Matching Error Estimation

At the second stage, we seek the best match between two fragments and calculate a corresponding matching error [5]. The two fragments are positioned in a way that two of their broken facets are facing each other (Fig. 3).

The matching error for each relative orientation is calculated by measuring the point by point difference of the curvature of the two surfaces, over a uniform grid. A measure based simply on the point by point distance of the two surfaces would not be reliable, as it would vary significantly even with small alterations of the surfaces' relative pose.

The matching error is measured using the depth buffer, to achieve uniform sampling. Imagine the depth buffer as a plane (u,v) placed between the two objects (Fig. 3a). The curvature at a surface point (u,v) can be regarded as the gradient of

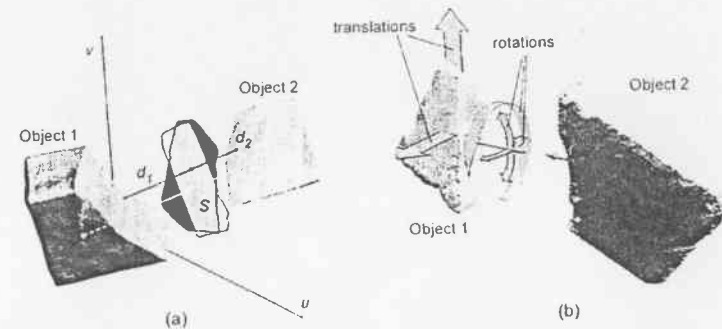


Figure 3. Partial fragment matching. (a) The object setup for the matching error calculation. (b) The error optimisation over the five degrees of freedom (2 translations and 3 rotations).

the depth buffer at this point, with regard to u and v .

The resulting matching error ϵ_J for two facets is:

$$\epsilon_J = \frac{1}{A_s} \iint_S \left(\left| \frac{\partial d_1(u,v)}{\partial u} + \frac{\partial d_2(u,v)}{\partial u} \right| + \left| \frac{\partial d_1(u,v)}{\partial v} + \frac{\partial d_2(u,v)}{\partial v} \right| \right) dS$$

where d_1 and d_2 are the depth buffers for the two surfaces, S is the buffer region where the two facets overlap and A_s is the corresponding area of overlap.

In order to find the optimal matching position, the matching error must be minimised over a set of parameters $\tilde{\omega}$ defining the two fragments' relative pose. To obtain all relative poses, one of the fragments is allowed to rotate around its centre of mass (three degrees of freedom) and slide along the broken facet (two degrees of freedom) (Fig. 3b).

Rotations and translations are limited to angles and displacements close to the original placement of the fragments. The object needs to perform a full circle rotation only around the axis that is perpendicular to the matching surfaces.

The error is minimised over $\tilde{\omega}$ using a global search method. In our system we

implemented Enhanced Simulated Annealing (ESA) [6], which produced good results (average optimum pose detection rate at 80%, while classic SA had a success rate of approximately 70%).

The ESA algorithm is a variation of the original Simulated Annealing method proposed by Kirkpatrick et al [7], which incorporates adaptive convergence and feature vector perturbation steps, as well as a search space partitioning strategy.

The material information, if available, can significantly improve the matching process. If two fragments consist of different raw materials (e.g. different types or age of marble), this pair would be discarded before entering the pairwise matching stage. Other textural information, such as the direction of marble veins can contribute to the matching process. The matching error can be weighted according to the deviation of the average vein direction vectors between the two fragments. The average direction vectors are estimated by measuring the vein directions in various locations on the marble object and interpolating the values appropriately [8].

Pairs can also be discarded due to other constraints, such as knowing that the two fragments belong to different larger

structures, or significant difference in the area of two broken facets.

The one-by-one matching stage can take several minutes, especially for large, ungrouped fragment sets. Therefore, this procedure is performed off-line and the minimum matching errors for every facet combination of all fragment pairs are stored in a look-up table (Fig 1). As the matching processes are mutually independent, this stage is highly parallel and trivially implementable on a distributed system.

5. The Full Reconstruction

For the final assembly of the fragments, we seek to minimise a global *reconstruction error* by choosing and clustering the appropriate combinations of fragments. The combination of object fragments is governed by a number of basic rules:

- i. A fragment can be attached, or *bound*, to as many other objects as the number of its facets labelled as broken.
- ii. The bond between two fragments is unique, i.e. only one facet of a fragment can be bound to another fragment.
- iii. Some fragments may not belong to a valid assembly.
- iv. Fragment pairs that yield smaller matching error must be favoured.

In our system, the globally optimised assembly of the fragments is performed by a kernel that generates and rearranges the bonds in real-time.

Every valid combination of fragment facets is assigned a fitness value that reflects the suitability of the fragment pair, which the facets belong to, to form a bond using these facets. A set of such combinations, which conform to the first two rules, forms a string that represents the currently active bonds among the fragments. This string is iteratively

mutated and its fitness values are adjusted until convergence is achieved.

At the beginning of the full reconstruction, all combinations are initialised with a fitness of 0.5 (fitness range is 0.0 to 1.0) and every fragment appears in the combination string at least once. The combinations with small pairwise matching error must be favoured (rule iv). For every object fragment, the combination with the smallest matching error it belongs to, whose facets have not been already used, is inserted in the string. The fitness of all initial pairs is increased.

The fitness of a pair is adjusted according to the global error that the current instance of the combination string produces. The global error is the sum of the matching error values of the individual combinations. If a string of combinations yields greater error than the previous one, the string is discarded and the fitness of all combinations is decreased, otherwise the fitness is increased.

Permutation of the combinations is performed by crossing-over the participants of some pairs of combinations. These combinations are selected among the less fit fragment pairs.

The algorithm terminates when the fitness values are stabilised within a small range for a sufficient number of iterations. Due to the simplicity of the algorithm and the precalculated error values for the pairs, the binding of the fragments is very fast and can be performed at interactive rates.

Figure 4 presents a four-step example of the algorithm's behaviour for a small set of four fragments with a total of seven facets. In step 1, the combinations are initialised and every pair's fitness is set to 0.5. Due to the fact that the fitness is 0.5 for all current pairs, two valid pairs are randomly selected and their participant facets mutually interchanged to form the combinations of step 2. The overall error of the second step is higher than the first, resulting in a decrease of the current combinations' fitness value. The algorithm then backtracks, preserving the new fitness

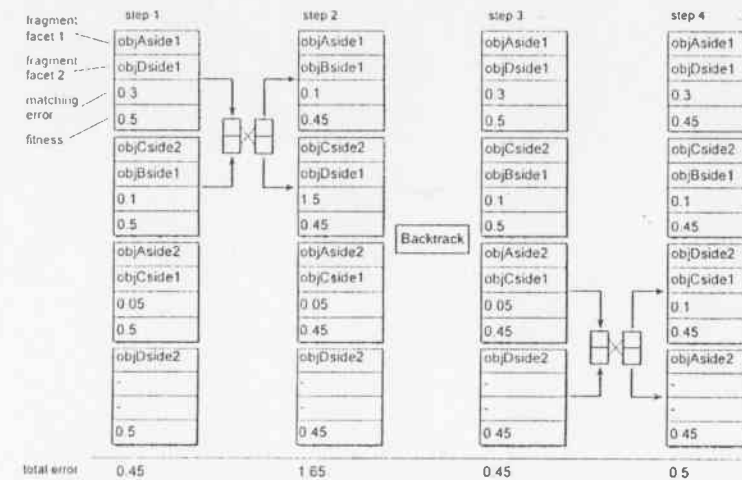


Figure 4. An example of the full reconstruction stage algorithm.

values. In step 3, the last two combinations are less fit and are therefore selected for the crossover.

If two objects belong to different structures and this fact is known a priori, the appropriate combinations would have been discarded at an early stage and therefore the combinations associated with these fragments would never appear in the combinations string. Apart from that, the user may bias the result by declaring via the user interface that two objects cannot connect to each other or must be glued together by force. The corresponding combinations will never or always appear in the combinations string, respectively.

6. System Implementation and Tests

The matching has been implemented in a modular way so that the off-line partial reconstruction can work independently on many machines simultaneously. The full

reconstruction is performed via a user interface that allows for the visualisation of fragment clustering and manual intervention in the process.

Due to the fact that the minimisation scheme of the one-by-one fragment matching stage requires that the matching error be estimated for hundreds of different fragment poses, the corresponding depth buffer maps of the facets to be matched must be acquired at a fast rate. Therefore, we used OpenGL for the hardware-accelerated rendering of the fragment facets and the acquisition of the depth maps. Typical one-by-one matching error estimation times are between 15 and 40 seconds for the average pair (5,000-15,000 polygons per fragment), while the full reconstruction stage requires only 0.3-2 seconds for 15 to 40 fragment sets.

We have tested our algorithm for the one-by-one matching with more than 50 fragments (both scanned objects and 3D models) at various rendering resolutions, from 32x32 to 256x256, depending on the mesh detail. Some of the original scanned

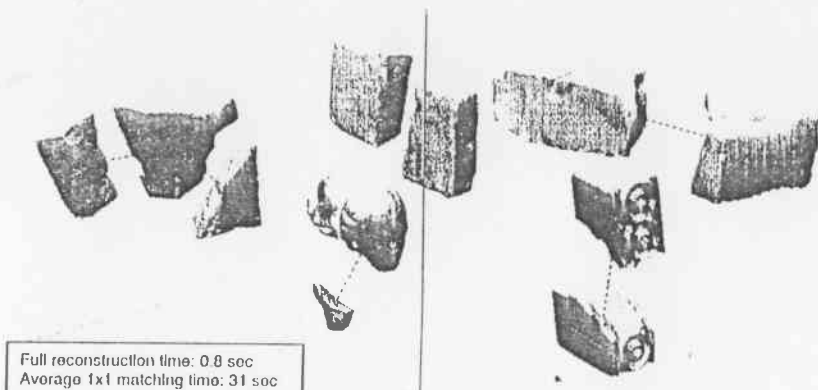


Figure 5. Full reconstruction of a test set comprising of objects of various types.

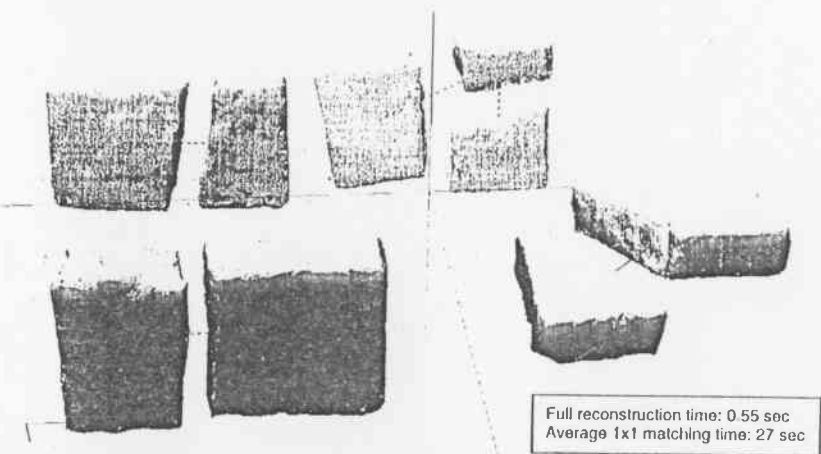


Figure 6. Full reconstruction of a test set. Links displayed as dashed lines are soft (computer generated) links while the normal lines (lower right pair) are hard (external) links.

pieces produced digitised meshes with up to 60,000 triangles per fragment. As most of these test objects were building blocks without decorative engravings, we could safely reduce the triangle count of most fragments below 10,000, thus speeding-up the matching process.

Running tests with the same pairs of fragments at various mesh resolutions

showed that the one-by-one matching success rate was sustained at about 80% after reasonable subsequent mesh simplifications. The loss of mesh detail increased the average number of iterations for the ESA convergence but this was counterbalanced by the decrease in the matching error calculation time due to better rendering performance.

The performance of the full reconstruction stage is reasonable. Unsupervised reconstruction (with neither constraints nor user intervention) resulted in correct reconstruction of more than 50% of the original objects. The performance can be significantly improved by imposing constraints (approximately 90% success rate), or resolving ambiguities by human intervention. In cases where many surfaces were similar to each other and the matching error variations small, some fragments failed to match. In those cases, it would be difficult to even manually match these fragments.

To allow for experimentation and manual intervention, fragments were allowed to be bound together with "hard" links. Hard links in contrast to the soft or algorithm generated links, are enforced by structural criteria or by manual linking via the interface. These links have the highest fitness and they are always included in the combination string and the final result. Figure 5 displays the results of an unsupervised reconstruction and figure 6 shows the final reconstruction for a small test set where both hard and soft links are used.

References

- [1] A.D. Kalvin, A. Remy, L.J. Castillo, K. Morla, E. Nolasco, J. Prado, V. Fernandez, R. Franco, and G. Wiese, "Computer-Aided Reconstruction of a pre-Inca Temple Ceiling in Peru," *Proc. Computer Applications in Archaeology (CAA 97)*, University of Birmingham, England, April 1997.
- [2] A.D. Kalvin, T.J. Watson, D. Dean, and J.J. Hublin, "Visualization Blackboard: Reconstruction of Human Fossils," *IEEE Computer Graphics and Applications*, vol. 15, no. 1, pp. 12-15, Jan. 1995.
- [3] R. Sablatng, C. Menard, and W. Kropatsch, "Classification of Archaeological Fragments Using a Description Language," *Proc. Eusipco 98*, vol. 2, pp. 1097-1100, Rhodes, Greece, 1998.
- [4] G. Papaioannou, E.A. Karabassi, T. Theoharis, "Segmentation and Surface Characterization of Arbitrary 3D Meshes for Object Reconstruction and Recognition", submitted to the International Conference on Pattern Recognition '2000.
- [5] G. Papaioannou, E. A. Karabassi, T. Theoharis, "Reconstruction of Three-dimensional Objects through matching of their Parts", submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [6] P. Siarry, G. Berthiau, F. Durbin, J. Haussy, "Enhanced Simulated Annealing for Globally Minimizing Functions of Many-Continuous Variables", *ACM Transactions on Mathematical Software*, 23(2), pp 209-228, 1997.
- [7] S. Kirkpatrick, C. D. Gelatt Jr., M. P. Vecchi, "Optimization by Simulated Annealing", *Science*, 220(4598), pp 671-680, 1983.
- [8] G. Papaioannou, T. Theoharis, A. Boehm, "A Texture Controller", *The Visual Computer*, Springer-Verlag, 14(10), pp 488-496, 1998.