Virtual Archaeologist: Assembling the Past

Georgios Papaioannou, Evaggelia-Aggeliki Karabassi, and Theoharis Theoharis
University of Athens

Reconstruction of archaeological monuments from fragments found at archaeological sites is a tedious task requiring many hours of work from archaeologists and restoration personnel.

For large constructions, such as the Parthenon at the Acropolis of Athens, the restoration process takes longer due to the mass of fragments. To test large building blocks against others for potential matching, sometimes archaeologists and site architects must move the cumbersome stones 50 meters or more away from their original locations using cranes. Fragments missing or deteriorated because of erosion or impact damage further hinder archaeological reconstruction.

Until now, computers have provided archaeologists tools for the digitization and archiving of artifacts,[1] visualization and virtual manipulation of 3D or 2D scanned objects, visual representation of historical sites through VR,[2] image processing, and restoration of frescos.[3] However, not much work has been conducted on automatic reconstruction of complete objects from arbitrary fragments.

Existing algorithms focus on the reconstruction of vases and rely either on classification of certain qualitative features of the fragments, as in Sablatnig et al.,[4] or on a comparison of the broken surface boundary curves to match and align the vase pieces.[5] The first method assumes that the structure of the final, complete object is known a priori and fragments are extensively labeled and categorized beforehand. The second completely disregards the interior of the broken surface and therefore is restricted to thin-walled objects.

Currently, the Digital Michelangelo team[1] is investigating approaches to assemble the Forma Urbis Romae—a marble map of ancient Rome—from 1,163 fragments. The team plans to face the problem as a jigsaw puzzle based on broken surface border signatures, while exploiting additional features of the fragments, such as thickness or marble veining.

Generally, the reconstruction of arbitrary objects from their fragments can be regarded as a 3D puzzle, taking into account the following considerations:

- Parts (fragments) have arbitrary shapes
- The shape and number of the final objects are unknown
- There's an arbitrary number of fractured faces per fragment
- Some fragments may be missing
- Surfaces are probably flawed or weathered
- No strict assemblage rules exist

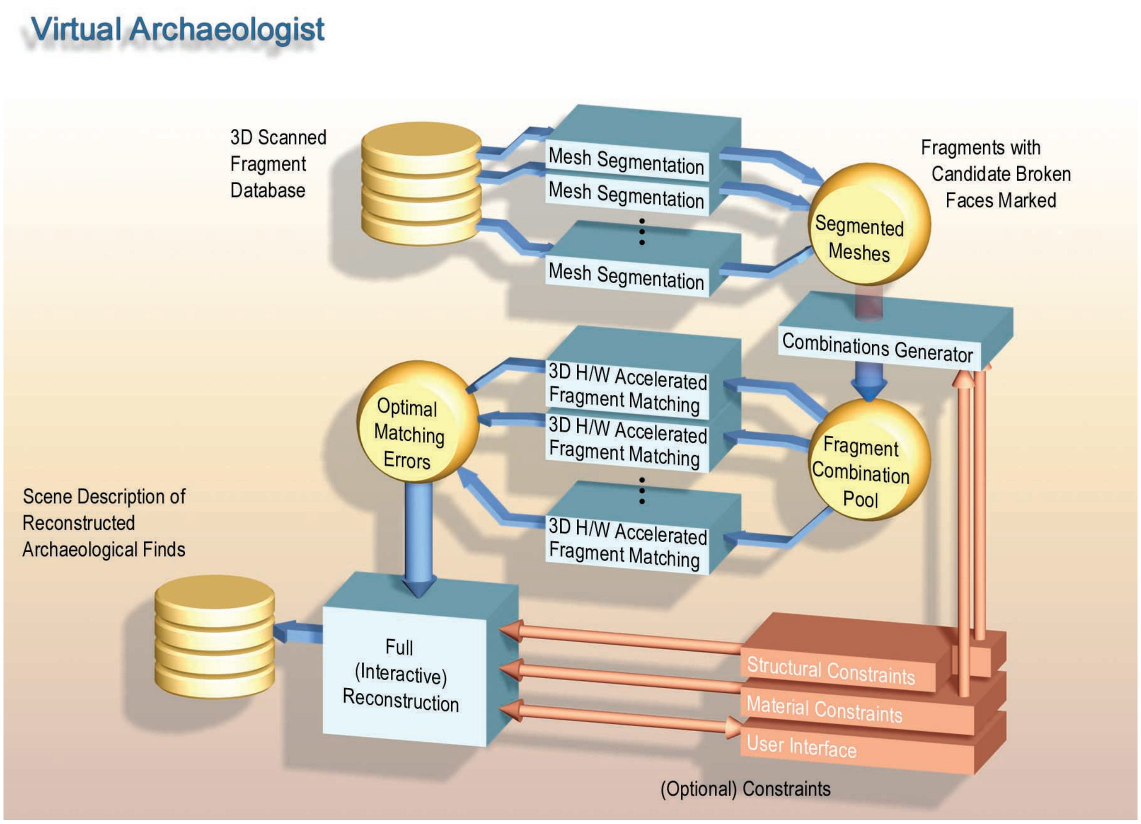
In this article we present a complete method—encapsulated in our Virtual Archaeologist system—for the full reconstruction of archaeological finds from 3D scanned fragments. Virtual Archaeologist is designed to assist archaeologists in reconstructing monuments or smaller finds by avoiding unnecessary manual experimentation with fragile and often heavy fragments. An automated procedure can't completely replace the archaeology expert, but provides a useful estimation of valid fragment combinations, and accurately measures fragment matches.

The Virtual Archaeologist system matches complementary scanned data, estimates the relative position of fragments, and clusters fragments belonging to the same entity.

In Virtual Archaeologist, we regard the reconstruction problem from a general, geometric point of view, relying on the broken surface morphology to determine correct matches between fragments. This approach doesn't require specific object information, but is versatile enough to exploit any other data available. A brief preliminary sketch of the underlying algorithms used in Virtual Archaeologist appears elsewhere.[6]

Our program detects candidate fractured faces, matches fragments one by one, and assembles (clusters) fragments into complete or partially complete entities. The only input data our system requires are the polygonal meshes of the original fragments. We acquired meshes with a 3D scanner or digitizer. Modeling or curve interpolation may be required in cases where only blueprints of cut sections of the fragments are available as

0272-1716/01/$10.00 © 2001 IEEE


Virtual Archaeologist
3D Scanned Fragment Database
Mesh Segmentation
Mesh Segmentation
Mesh Segmentation
Fragments with Candidate Broken Faces Marked
Segmented Meshes
Combinations Generator
3D H/W Accelerated Fragment Matching
3D H/W Accelerated Fragment Matching
3D H/W Accelerated Fragment Matching
Optimal Matching Errors
Fragment Combination Pool
Scene Description of Reconstructed Archaeological Finds
Full (Interactive) Reconstruction
Structural Constraints
Material Constraints
User Interface
(Optional) Constraints

1 Virtual Archaeologist architecture and data flow.

part of the standard archiving procedure. An optional set of constraints—such as material or structural fragment attributes—significantly improves the overall accuracy and performance. Our system doesn't require human intervention, but users can clarify the reconstruction by interactively fine-tuning the clustering and poses of the fragments.

Method overview

We divide the reconstruction of an object into three stages (Figure 1). The first stage is the mesh segmentation. To minimize the search space when "gluing" two fragments together, we restrict matching between potentially "interesting" sides of the fragments. Based on the observation that fractured surfaces—even weathered ones—tend to be more rough and jagged, we segment each mesh into facets and only mark as candidates for matching those facets that exhibit relatively high coarseness.

In the second stage (fragment matching), after generating all valid fragment-pair combinations, we estimate the relative pose for all fragment pairs and all candidate facets with minimized matching errors. We examine each pair of fragments to determine the relative orientation that corresponds to the best fit and therefore the minimal matching error per facet pair. Note that not every possible pair of fragments enters this process, as many are discarded because of incompatible material or target structure, if such information is available (see the optional constraints in Figure 1).

The matching error estimator uses hardware accelerated 3D rendering of the fragments and operates on the depth buffers produced. Our application uses rendering not only for visualization, but for the matching process as well.

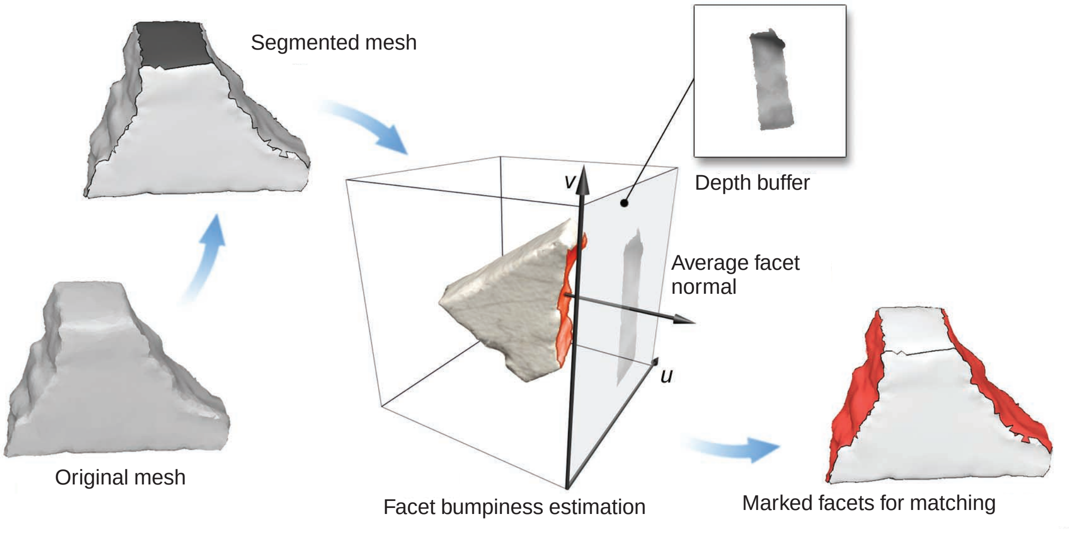
When the system calculates all optimal pairwise matching error values and stores them in a table (fragment facet by fragment facet), the third stage (full reconstruction) selects fragment combinations that minimize a global reconstruction error. This reconstruction error equals the sum of matching errors of a given set of fragment pairs.

External constraints may contribute to this stage as well, reducing the time needed to produce a correct fragment clustering by eliminating a large number of combinations.

Both the mesh segmentation and fragment matching stages occur offline. Each time we add a new fragment mesh to the object database, the system segments and calculates and stores the matching error values. These procedures are not interactive because although the segmentation of a mesh lasts no more than 2 seconds, fragment matching can take up to 30 seconds per facet pair. Note, we measured all times on a 450-MHz Pentium III equipped with a 185-MHz TNT2 Ultra graphics accelerator.

Mesh segmentation

Our system first partitions a fragment mesh into areas of adjacent nearly coplanar polygons, corresponding to the object's facets (Figure 2). This surface segmentation process works using a simple region-growing algorithm.[7] The process begins with an arbitrary polygon. Neighboring polygons are classified to the same region if their normals don't deviate from the average region



2 Mesh segmentation and detection of broken sides.

normal by more than a predefined threshold; otherwise a new region forms.

During the growing process, small surface regions may be created within larger ones. Since we must partition the mesh into "crude" facets, we apply a merging stage to clean up the facets and eliminate small erroneous regions. We consider a region "small" if it covers less than 5 percent of the entire mesh surface area. Insignificant regions are eliminated by iteratively assigning the polygons of small surface areas to large adjacent regions.

Having partitioned the fragment mesh into crude facets, we label and group those facets that exhibit higher coarseness. We can't accurately measure a facet's bumpiness from the original mesh (unless the surface is uniformly sampled) because each facet consists of polygons of arbitrary connectivity and area. Instead, we use an image-based bumpiness measure calculated on the facet's elevation map.

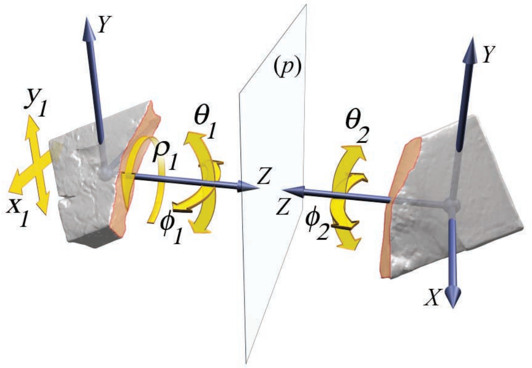
The elevation map is essentially a 2D array—representing the distance of the facet from a plane perpendicular to the average facet normal—measured at equidistant grid points. The map equals the contents of the depth buffer when the facet triangles are rendered with the viewing direction parallel to the average facet normal (Figure 2).

A surface's bumpiness is associated with the rate of elevation variance. It can be estimated effectively on the elevation map with an image filter, such as the Laplace image operator.

Since engraved sides of a fragment are inherently bumpy, they are marked as well. This isn't a problem, however, because these facets are incompatible with any other and will produce a high matching error during the next stage. On the contrary, if the broken sides are smooth, they may not be marked automatically and thus require manual selection.

Fragment matching

To see if two fragments are complementary, we seek the best match between them with respect to their relative pose and calculate corresponding matching errors.



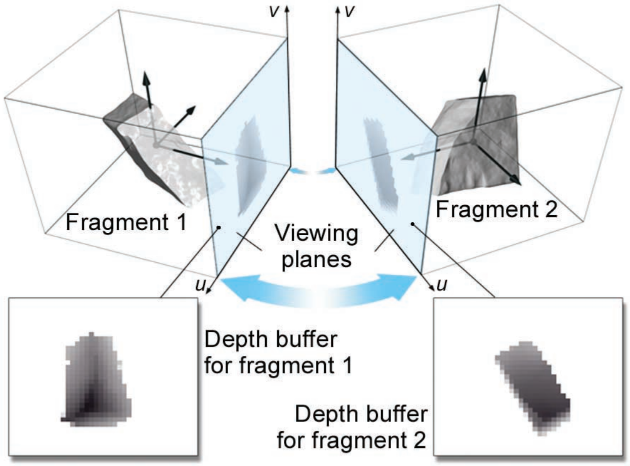
3 Fragment pose parameters for the matching error calculation.

This process repeats for all marked facet pairs of all fragment combinations.

First, we position two fragments so that two of their broken facets face each other (Figure 3)—for example, if we align the average facet normal. A set of seven pose parameters adequately represents the two fragments' alignment. More specifically, the first object can perform a full rotation around the axis of alignment (ρ_1), deviate from the axis (ϕ_1, θ_1) by up to 10 degrees, and slide along the broken facet (x_1, y_1). The ability to slide is essential in locating potential partial matches between facets of different size or shape. The second object need only diverge from the axis of alignment by up to 10 degrees (ϕ_2, θ_2). Although the last rotations are theoretically redundant, they improve the convergence of the search for optimal matching.

Ideally, if the fragments' broken surfaces are complementary, the matching error should be zero for a relative pose of the two pieces where they "fit" together. For all other placement configurations or for incompatible fragments, the matching error between the fragment facets is significant.

A naive way to estimate the matching error for each set of pose parameters would be to sum all point-to-point distances between corresponding points on the facing surfaces of the two fragments. Unfortunately, because this solution depends on the closest distance



4 **Point-to-point distance calculation with the depth buffer.**

between the two fragments, it's sensitive to noisy data. Worse, small variations of the pose parameters produce drastic changes to the resulting error, making the measure unreliable and difficult to optimize.

Instead of using the point-to-point distances directly, we work on their derivatives—that is, curvature of the surfaces measured from a uniform grid on a plane (p). The system positions (p) between the fragments so that it's perpendicular to the original average normal axes (Figure 3). The resulting matching error ε_d for two facets is

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

where d_1 and d_2 are the distances of the two surfaces from the separating plane (p), S is the region of overlap between the surface projections on (p), and A_S is the corresponding area of overlap. Small, isolated surface flaws or sampling errors have a local effect on the error because of its differential form, so the method tolerates noisy input data.

In practice, we measure the matching error using the depth buffer. Imagine an observer looking at the two fragments through a viewing plane coincident with the separating plane (p), as illustrated in Figure 4. Rendering each fragment separately, we obtain two depth buffers $Z_1(i, j)$ and $Z_2(i, j)$, $i=1, \ldots, N_u, j=1\ldots N_v$. Surface curvature for corresponding points on the fragments is uniformly sampled and easily obtained from the derivatives of these two buffers with respect to u and v. The partial derivatives of the continuous error measure are replaced by forward differences on the depth buffers:

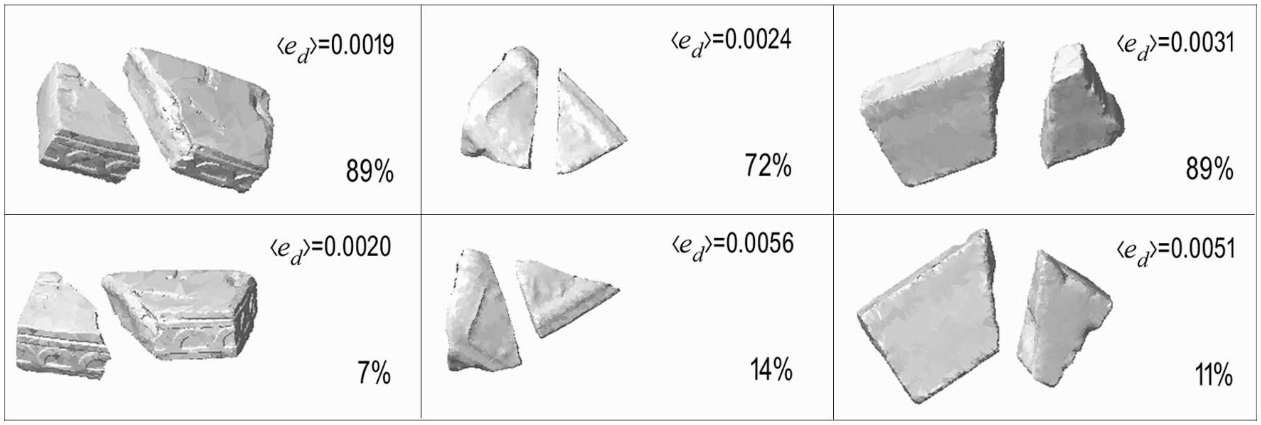
$$\Delta_u Z(i,j) = Z(i+1,j) - Z(i,j)$$
$$\Delta_v Z(i,j) = Z(i,j+1) - Z(i,j)$$

The matching error is evaluated as

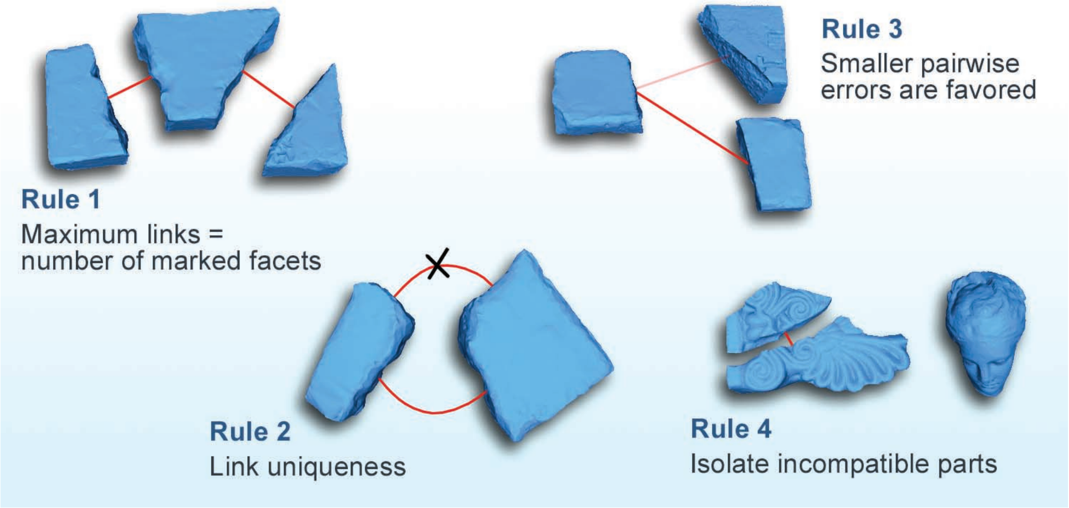
$$\varepsilon_d = \frac{1}{A_S}\sum_{(i,j)\in S}\left(|\Delta_u Z_1(i,j)+\Delta_u Z_2(i,j)| + |\Delta_v Z_1(i,j)+\Delta_v Z_2(i,j)|\right)$$

In the above equation, S is the set of depth buffer cells where both depth buffers have noninfinite values and A_S is the number of elements in S.

We use a global optimization method to minimize the error over the set of pose parameters. In our system, we implemented enhanced SA, a variation of simulated annealing (SA),[8] which produces good results (the average optimum pose detection rate is 85 percent). The details of SA are beyond the scope of this article, but you can refer to any combinatorial optimization or search methods book for details on SA, as well as to Siarry et al.[9] for the enhanced SA method we adopted for our application. Figure 5 shows some representative examples.



5 **Fragment matching examples for scanned objects. The top row presents correct matches along with the corresponding error measures and success rates. The bottom row shows the most frequent erroneous results.**


Rule 1
Maximum links = number of marked facets
Rule 2
Link uniqueness
Rule 3
Smaller pairwise errors are favored
Rule 4
Isolate incompatible parts

6 **Full reconstruction stage assemblage rules.**

Full reconstruction

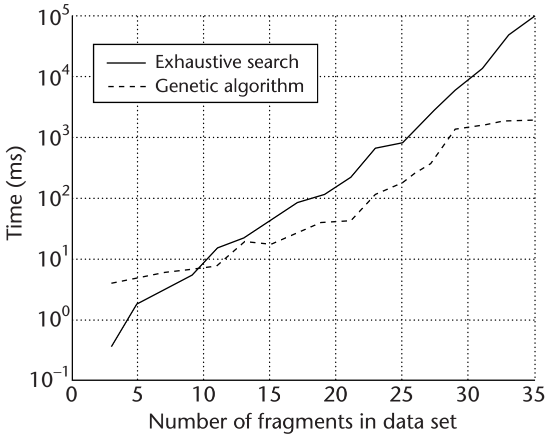
In the final reconstruction, we try minimizing the global *reconstruction error*, (which is the sum of the matching errors of the individual combinations currently active). Four principal rules govern the fragment assemblage (Figure 6):

1. A fragment can be linked to as many other objects as the number of its facets that are marked as broken (for example, if a fragment has *N* facets marked as broken, then it can be linked to *N* other objects at most).
2. The bond between two fragments is unique.
3. Fragment pairs that yield a smaller matching error must be favored.
4. Fragments may exist that don't belong to a valid reconstructed object and must be isolated.

A link between two fragments corresponds to a combination of fragment facets whose matching error and relative pose have been precalculated. A kernel that generates and rearranges the links in subsecond time forms the globally optimized fragment assemblage. This is an important feature, because the user can experiment with the fragments and shape the final result interactively by explicitly joining or separating them.

If the number of fragments in the data set is fairly small (less than 30 pieces), the set of fragment combinations that yields the smallest reconstruction error is determined using an exhaustive search. Since an exhaustive search leads to an exponential increase of execution time, we use a genetic algorithm for large data sets to reduce the number of iterations needed (Figure 7).

In the case of the genetic algorithm, the currently active set of combinations is repeatedly mutated by crossing-over combinations. The new set of combinations may be accepted or rejected according to the new reconstruction error. The algorithm terminates when a set of combinations is fit (has prevailed long enough). More details on the specific algorithm may be found in Papaioannou et al.[6]



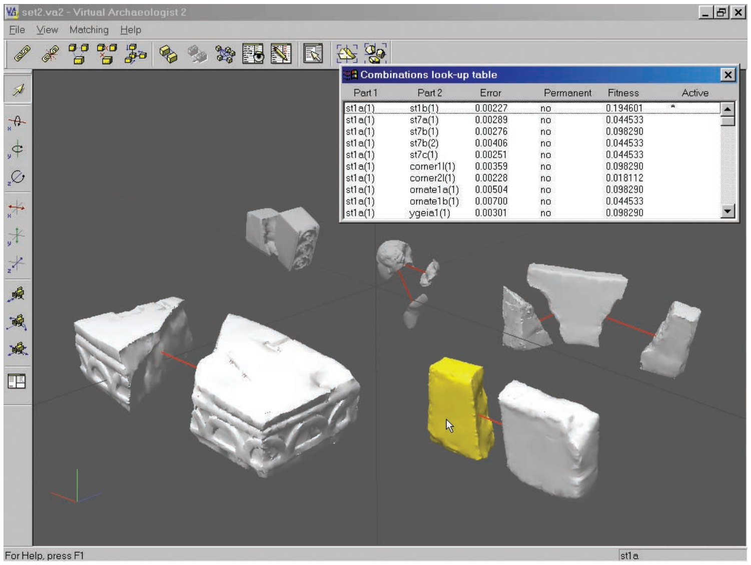
7 **Full reconstruction stage execution time.**

System implementation

We implement Virtual Archaeologist in three modules, one for each stage, so that they can work independently.

We perform the preprocessing once, after the polygonal data acquisition, and the segmentation information is stored in the mesh file. We use OpenGL for the calculation of the facet elevation maps. The segmentation lasts no more than 2 seconds even for complex models (approximately 90,000 triangles) for 128 × 128 depth buffer resolution.

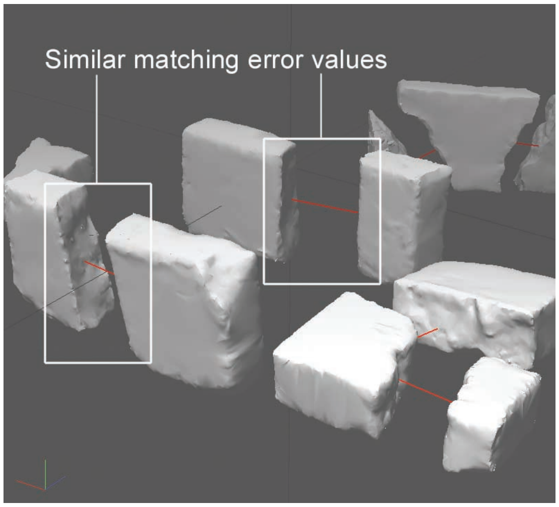
The fragment matching module uses hardware accelerated OpenGL rendering for the error estimation, achieving an average of 10 matching error samples per second. Depth buffer resolution can be adjusted to trade accuracy for computation speed. Typical dimensions for the depth buffer in this application are between 100 × 100 and 256 × 256 pixels. Matching lasts between 15 and 40 seconds per facet pair, depending on the mesh triangle count. Because fragment matching is computationally expensive, the system incrementally invokes updates to the file of matching errors when a new piece is added to the collection.



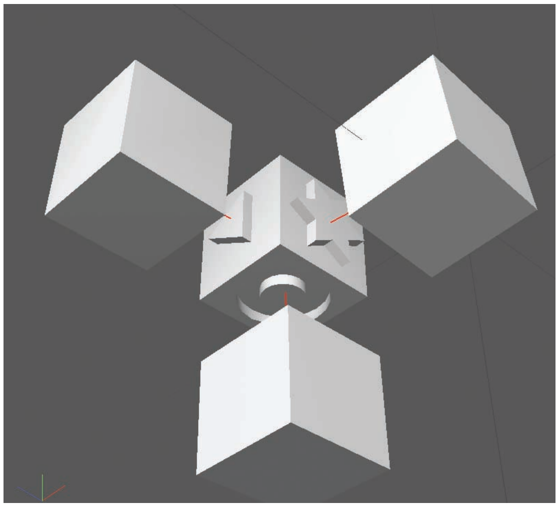
Combinations look-up table

Part 1	Part 2	Error	Permanent	Fitness	Active
st1a(1)	st1b(1)	0.00227	no	0.194601	*
st1a(1)	st7a(1)	0.00289	no	0.044533	
st1a(1)	st7b(1)	0.00276	no	0.098290	
st1a(1)	st7b(2)	0.00406	no	0.044533	
st1a(1)	st7c(1)	0.00251	no	0.044533	
st1a(1)	corner1l(1)	0.00359	no	0.098290	
st1a(1)	corner2l(1)	0.00228	no	0.018112	
st1a(1)	ornate1a(1)	0.00504	no	0.098290	
st1a(1)	ornate1b(1)	0.00700	no	0.044533	
st1a(1)	ygeia1(1)	0.00301	no	0.098290	

8 The Virtual Archaeologist desktop in action.



9 Reconstruction example with possible ambiguities.



10 Simple 3D puzzle—a nonarchaeological case study.

The interactive full reconstruction is performed in a 3D environment, as depicted in Figure 8. This lets the user edit the final result by providing standard manipulation functionality (fragment position and rotation), and additional tools via the user interface. These tools include linking and separating fragments, measuring the matching error in real-time between two fragments, and calculating the poses for the reconstructed objects automatically.

Case studies

Range scanners have been used in field archaeology only recently. Archaeological sites, such as the Athens Acropolis, don't yet have the necessary equipment to create databases of scanned finds. We've tested Virtual Archaeologist with 3D-digitized plaster scale models of objects (mostly building block replicas and ceramic pot fragments).

The majority of the fragments weren't well preserved and the broken surfaces were smoothed out. Even on sharp-edged pieces, we deliberately chipped off protrusions during the experiments to simulate deterioration. In general, we had no perfectly matching fragments, which is common in all archaeological finds. The fragment collections we used ranged from 3 to 35 pieces.

Virtual Archaeologist (with no user intervention) correctly reconstructed 90 percent of the original objects. We used the broken surface area similarity as an effective constraint in the final reconstruction phase. If the area of the broken facets differed by more than 20 percent, the system prohibited this combination. The existence and the parameters of this constraint can be adjusted from the user interface.

In experiments where many surfaces were similar and the matching error variations were small, some fragments failed to match. In these instances, however, it was sometimes difficult even for a human to match these fragments (Figure 9). Moreover, since the automatic assembling algorithm doesn't have knowledge about the expected shape of the reconstructed object, in some cases the fragment combinations that minimized the matching error didn't correspond to valid objects. Manual linking of the fragments resolved ambiguities during the interactive reconstruction stage.

To demonstrate our system's ability to assemble in 3D based only on geometric information and without any constraints, we produced a small 3D puzzle (Figure 10) with a modeling package. Virtual Archaeologist matched the pieces perfectly.

Until now, we have tested the Virtual Archaeologist extensively with artificial objects and replicas of real artifacts. We currently are discussing its introduction to archaeological sites, starting with small collections of marble fragments from the Acropolis site. We also are working towards the integration of our matching environment in a larger system that will include other multimedia materials and specific technical fragment details. ■

Acknowledgments

This work was partially funded by the National and Kapodistrian University of Athens, Research Grant 70-4-3241.

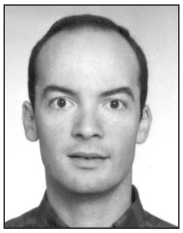
We'd like to thank Nikolaos Toganidis (Chief Architect, Parthenon Restoration Project) for motivating this work.

References

1. M. Levoy et al., "The Digital Michelangelo Project: 3D Scanning of Large Statues," *Computer Graphics* (Proc. Siggraph 2000), ACM Press, New York, July 2000, pp. 131-144.
2. E. Berndt and J.C. Teixeira, "Cultural Heritage in the Mature Era of Computer Graphics," *IEEE Computer Graphics and Applications*, vol. 20, no. 1, Jan./Feb. 2000, pp. 36-37.
3. A.D. Kalvin et al., "Computer-Aided Reconstruction of a pre-Inca Temple Ceiling in Peru," *Proc. Computer Applications in Archaeology* (CAA 97), Apr. 1997.
4. R. Sablatnig, C. Menard, and W. Kropatsch, "Classification of Archaeological Fragments Using a Description Language," *Proc. European Association for Signal Processing* (Eusipco 98), vol. 2, 1998, pp. 1097-1100.
5. G. Ucoluk and I.H. Toroslu, "Automatic Reconstruction of Broken 3D Surface Objects," *Computers and Graphics*, vol. 23, no. 4, Aug. 1999, pp. 573-582.
6. G. Papaioannou, E.A. Karabassi, and T. Theoharis, "Automatic Reconstruction of Archaeological Finds—A Graphics Approach," *Proc. Int'l Conf. Computer Graphics and Artificial Intelligence 2000*, 2000, pp. 117-125.
7. G. Papaioannou, E.A. Karabassi, and T. Theoharis, "Segmentation and Surface Characterization of Arbitrary 3D Meshes for Object Reconstruction and Recognition," *Proc. Int'l Conf. Pattern Recognition 2000*, IEEE Computer Soc. Press, Los Alamitos, Calif., 2000, pp. 734-737.
8. S. Kirkpatrick, C.D. Gelatt, Jr., and M.P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, 1983, pp. 671-680.
9. P. Siarry et al., "Enhanced Simulated Annealing for Globally Minimizing Functions of Many-Continuous Variables," *ACM Trans. Math. Software*, vol. 23, no. 2, 1997, pp. 209-228.

Georgios Papaioannou *is a member of the Computer Graphics Lab at the Department of Informatics and Telecommunications of the University of Athens, Greece. He received his BSc degree in computer science from the University of Athens in 1996. His main research interests are 3D modeling and rendering, as well as computer vision and the application of optimization techniques on computer graphics. He is currently working on his PhD thesis on the reconstruction of 3D objects from fragments. He is a member of the IEEE.*

Evaggelia Aggeliki Karabassi *is a member of the Computer Graphics Lab of the Department of Informatics and Telecommunications of the University of Athens, Greece. Her research interests include 3D modeling, collision detection, rendering, and 3D signal analysis. She received her BSc and PhD degrees in computer science from the University of Athens in 1996 and 2001, respectively. She is a member of the IEEE.*



Theoharis Theoharis *is an assistant professor in the Department of Informatics and Telecommunications of the University of Athens, Greece and is head of the Computer Graphics Lab. His main research interests are computer graphics and parallel processing. He received his PhD degree in computer graphics and parallel processing from the University of Oxford, England.*

Readers may contact Papaioannou at the University of Athens, Greece, email georgep@di.uoa.gr.