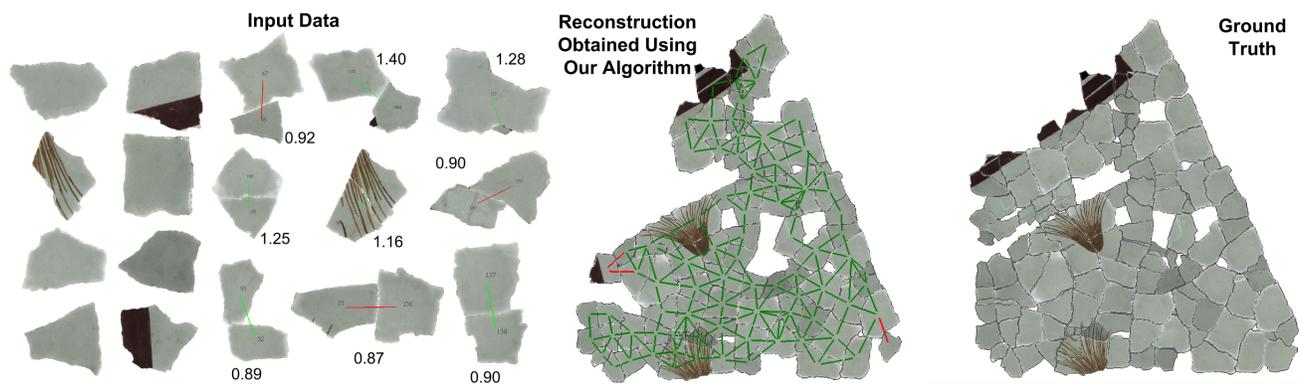


Wall Painting Reconstruction Using a Genetic Algorithm

Elena Sizikova and Thomas Funkhouser

Department of Computer Science, Princeton University, USA



Abstract

Global reconstruction of two-dimensional wall paintings (frescoes) from fragments is an important problem for many archaeological sites. The goal is to find the global position and rotation for each fragment so that all fragments jointly "reconstruct" the original surface (i.e., solve the puzzle). Manual fragment placement is difficult and time-consuming, especially when fragments are irregularly shaped and uncolored. Systems have been proposed to first acquire 3D surface scans of the fragments and then use computer algorithms to solve the reconstruction problem. These systems work well for small test cases and for puzzles with distinctive features, but fail for larger reconstructions of real wall paintings with eroded and missing fragments due to the complexity of the reconstruction search space. We address the search problem with an unsupervised genetic algorithm (GA): we evolve a pool of partial reconstructions that grow through recombination and selection over the course of generations. We introduce a novel algorithm for combining partial reconstructions that is robust to noise and outliers, and we provide a new selection procedure that balances fitness and diversity in the population. In experiments with a benchmark dataset our algorithm is able to achieve larger and more accurate global reconstructions than previous automatic algorithms.

Categories and Subject Descriptors (according to ACM CCS): 2D Reconstruction, Genetic Programming, Machine Learning, Statistics, Computational Archaeology, Data Mining, Machine Learning

1. Introduction

At archaeological sites around the world, many artifacts are found shattered into small fragments, and archaeologists face a difficult reconstruction task of reassembling the fragments, similarly to how one would arrange pieces of a jigsaw puzzle. For example, the ancient Greek civilization of Thera in the Akrotiri settlement created many wall mosaic paintings more than 3,500 years ago that have been left buried under volcanic ashes until today, when reconstruction of these wall paintings provides valuable information about the ancient civilization [Dou92]. Reconstruction

of fragmented wall paintings is part of a more general and challenging problem of 2D and 3D reconstruction of fragmented objects: torn documents [BBB05], archaeological artefacts [KL06], puzzles [SDN13], as well as other computer vision and graphics tasks [LK*81, RL01, HKH*10], where viewing fragments as part of an arrangement provides a more insightful holistic understanding.

Finding a globally consistent arrangement of fragments in an artifact is a notoriously difficult problem [BLD*12]. Fragments are fragile, present in large quantities, and cumbersome to han-

dle, which makes reconstruction systems difficult for manual effort even for relatively small-sized artefacts [WC08]. Computer-based methods have been developed in which fragments are scanned to produce 3D meshes and then reconstruction is performed automatically [BTFN*08, CBR*11] or with moderate human guidance [CWA*01, LGJ*07]. However, those methods still are far from being able to reconstruct large artifacts with missing fragments and erosion, which makes them impractical for typical large-scale archaeological reconstruction scenarios.

To make the computer-modeled reconstruction more practical, it is typically sub-divided into two sub-problems: local assembly, which handles finding which fragments should be adjacent, and global assembly, which handles finding globally optimal fragment placements [HFG*06]. We focus on the global assembly problem. Given a set of mostly incorrect pairwise fragment constraints on position and orientation (matches) predicted by a local assembly algorithm, we aim to find the largest set of matches that consistently and correctly places fragments into a common coordinate system (solve the puzzle).

There exist methods that tackle the specific problem of wall painting reconstruction [CBR*11]. While these methods can tolerate a small percentage of incorrect matches, they generally are unable to reconstruct large wall paintings because they are greedy, and therefore not robust to noisy and incorrect matches or incorrect partial reconstructions.

We introduce the idea of using genetic algorithms in archaeological reconstruction. To formulate wall painting reconstruction as a problem that can be solved by evolutionary computation, we describe a selection procedure that filters a population of partial reconstructions and a recombination procedure that generates new reconstructions from existing candidates. We use these procedures to perform a robust search of the large space of fragment reconstructions (which describe the placement of a subset of fragments). Our selection procedure uses a novel fitness function that mathematically captures the characteristics of a correct partial reconstructions. Our recombination procedures uses an Iterative Normalized Eigenvector Method that is robust to outliers. The combination is robust to incorrect matches and thus can build larger correct reconstructions than previously possible.

The contributions of our work are:

- A genetic algorithm framework for wall painting reconstruction that creates larger and more consistent reconstructions than was possible previously.
- A novel robust Iterative Normalized Eigenvector Method (it. NEVM) that handles noisy matches and discard outliers.
- A cluster selection procedure that encourages selection of clusters with potential for growth.

2. Related Work

The problem of fragmented object reconstruction is a classical problem faced in various fields: puzzle solving [WSKL88], re-assembling fragmented artefacts [WSKL88], and reconstructing torn documents [BBB05]. The idea of using an automatic approach for this task dates back to [FG64]. [KS09] and [WC08] provide

good overviews of recent methods considered in assembling broken 2-dimensional and 3-dimensional objects. Previous work can be roughly subdivided into methods of local and global assembly.

Local Assembly

Methods that address the local assembly problem typically focus on generating a set of pairwise matches between fragments. These methods consider properties that would be shared by adjacent fragments. For example, fragments could be matched based on face image similarity (when available) [URD*04, FT05, SE06], 2D fragment contour complementarity [KK01, DGLS02, PPE*02], or fracture surface alignment [BTFN*08, VB12, VVSB14]. Supervised learning approaches that aggregate many properties have also been used to predict fragment matches [TFBW*10, FSTF*11]. These methods usually produce large sets of potential matches, most of which are incorrect. Our goal is to select the maximal subset of correct matches to form a consistent global assembly from predicted matches.

Global Assembly

Global assembly methods have been proposed to select a subset of predicted matches that place fragments consistently within a common coordinate system. These methods need to address several mutually-related challenges: how to detect match outliers or noisy transformations, and how to navigate the exponential search space of the possible fragment arrangements.

Reconstruction of fragmented objects can be viewed as a graph [PK03, SBVV14], in which fragments are represented by nodes and matches are represented by edges. To navigate the search space, various heuristic approaches are usually applied. Agglomerative hierarchical clustering methods [BK93, ÜT99, PPE*02, BBB05, KK01, MP06, DS09, SE10, CBR*11, SBVV14] start by initializing individual fragments as clusters, and proceed by merging clusters, where merges can be ranked by a scoring function, until all fragments are contained in one cluster or no new merges can be made. In dense cluster growth methods [KK01, GMB02, MK03], a best cluster (or small set of best clusters) is incrementally grown until a heuristic stopping criterion is reached. A limitation of these approaches is that they are greedy: a promising but incorrect merge between partial reconstructions early in the process will be propagated until the end of the algorithm, without the possibility of being corrected.

To reduce the possibility of selecting incorrect merge configurations, some authors apply strict conditions on when a merge is created. For example, [GMB02] proposed a reconstruction approach when a fragment is added to a cluster only when it could be linked by at least two matches. While this condition can avoid many false positives, it is detrimental if fragments are sparsely connected to each other.

Genetic Algorithms

Genetic algorithms (GA) [Joh92] allow exploration of multiple partial reconstructions at once, and avoid early selection of a single

solution. GAs have previously been applied to solving jigsaw puzzles [TFSM02]. Recently, [SDN13] provided an overview of existing approaches, and reconstructed 22,834-piece puzzles. While wall painting reconstruction can be seen as an instance of solving a jigsaw puzzle, there are several fundamental differences that distinguish this specific problem. Jigsaw puzzle borders are typically known a priori, while in wall painting datasets the edges of the puzzles are most often unknown. Next, jigsaw puzzle pieces exhibit much less shape variation than aged, weathered artifact fragments, some of which may be missing, causing artificial holes in the reconstruction. Our approach to wall painting reconstruction is to model it as an evolutionary problem and design selection and recombination operations to tackle its specific challenges.

3. Approach

In this paper, we introduce a novel fully automatic method for reconstruction of fragmented wall paintings.

We propose a genetic algorithm (GA) for wall painting reconstruction. Genetic algorithms have been previously shown to be able to successfully navigate large multi-dimensional and unstructured search spaces that are challenging for traditional optimization techniques [Joh92]. We show how this approach can benefit reconstruction of partial and fragmented wall paintings.

The design principles of GAs are typically guided by a principle of survival of the fittest in a diverse population of individuals that are allowed to mate (merge). In our algorithm, clusters are iteratively re-combined using a probabilistic procedure that considers merges that will likely result in best new candidates, while the parent clusters are kept in the population. A novel robust convex optimization method is applied to each new cluster that iteratively discards outliers and creates a consistent set among the pairwise matches. To handle the large search space, a selection procedure is introduced that ranks clusters and allows fair selection of the top candidates. During the course of the selection procedure, clusters are compared using a fitness function that highly ranks clusters with characteristics that allow growth.

The key idea behind the procedure design choices is that it is difficult to distinguish correctness of matches and clusters without assembling them into larger entities and evaluating their consistency. While [CBR*11] applied this thought to evaluation of potential merges, we show that by incorporating it into all procedures of the algorithm while simultaneously removing reliance on greedy search yields better solutions overall.

4. Algorithm

The input to our algorithm is a set of predicted pairwise matches between fragments. Each match is represented by a transformation that aligns the fragment boundaries and a numerical score that ranks it relative to others [FSTF*11], so that higher scoring matches are more likely to be correct than lower scoring matches. The output is a reconstruction – a cluster of fragments connected by a consistent set of matches.

The algorithm proceeds as shown in Figure 1. It starts with an initial population of clusters consisting of one or two fragments.

The clusters then pass through a number of evolutionary iterations consisting of selection and recombination that create larger clusters. Evolution terminates either when a fixed number of iterations have occurred, or when no new clusters can be generated. The following section describes these steps in detail.

4.1. Initialization

The first step of the algorithm is to create an initial set of clusters. In our implementation, the initial clusters consist of singleton clusters, consisting of single fragments and no matches, as well as pair clusters, consisting of two fragments and a match. Pair clusters are generated directly by the pairwise matches provided as input.

4.2. Selection

At the start of each iteration, it is necessary to select from a large population of similar clusters only the clusters that will meaningfully contribute to evolution. To perform selection, we employ a problem-specific fitness function (See Fitness Function section for more details) to rank the clusters in the current population. We sought a procedure that would achieve three goals: separate correct clusters from wrong, rank clusters by size and preserve fragment diversity in the population. We thus perform two sub-selection procedures. The first is a diversification step, where clusters in the population that have a high fragment Jaccard coefficient J (number of shared fragments divided by total number of unique fragments) are filtered so that only the more fit cluster is kept. To make the process of comparing every pair of clusters efficient, clusters are assigned hash values of binary strings that represent the fragments they contain. The second is a variant of rank-selection, where clusters are binned according to $B = 114$ bins, where each bin represents a fragment that is part of the cluster (a cluster can be part of multiple bins). We then select the top clusters from each bin until a selection threshold S has been satisfied.

4.3. Recombination

After the best subset of clusters has been identified, it is used to generate new clusters. Recombination (also known as crossover) is the process of creating a new cluster given two parent clusters. In our case, the parent clusters need to share either a common fragment (Recombination By Fragment) or a set of spanning matches (Recombination By Match), so that the aligning transformation can be estimated. The recombination step is run once in each iteration.

Recombination By Fragment

Given a pair of clusters that share a fragment, an aligning transformation is specified which defines a potential child cluster. Since different fragments may induce different spanning transformations, we consider all possible spanning fragments (and induced transformations), and choose the highest scoring resulting child cluster to be created.

Recombination By Match

A configuration in which two clusters and a spanning match between a fragment from the first cluster and a fragment from the sec-

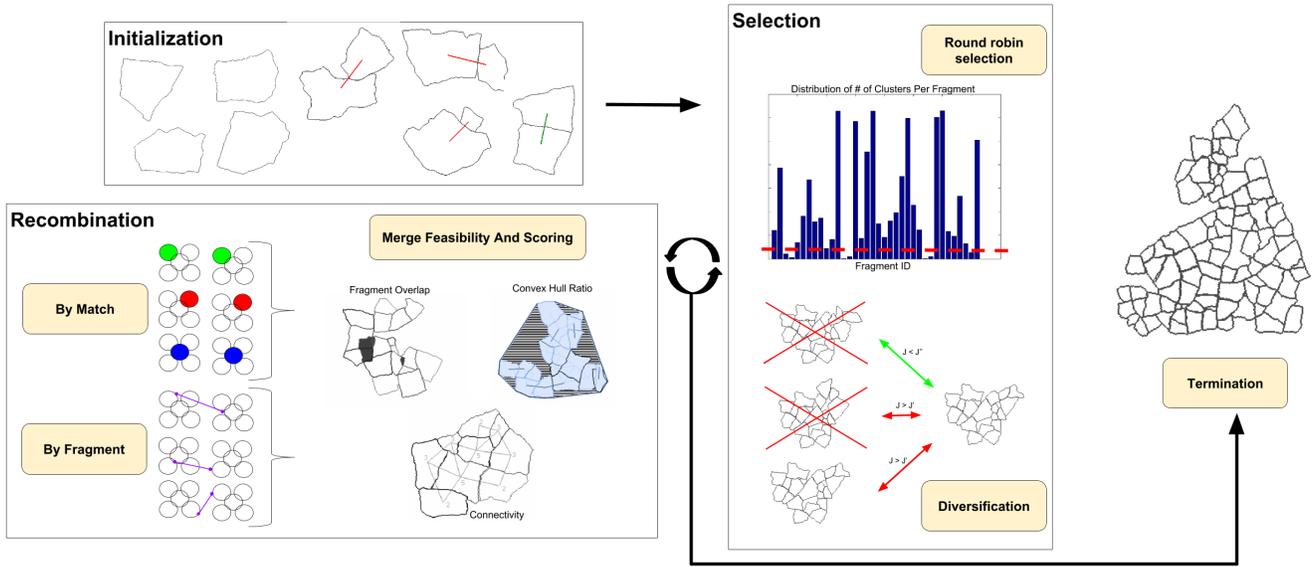


Figure 1: Overview of the Genetic Algorithm (GA) pipeline. The main steps of the algorithm are: Initialization, Selection and Recombination. The last two steps are iterated until a terminal criterion: a fixed number of iterations that have happened, or until no new clusters can be created (convergence).

ond cluster is given uniquely specifies a potential new child cluster. Since the space of spanning matches is typically very large, we employ a roulette-sampling procedure (weighted sampling with replacement) to choose which matches will be considered, with matches that have a higher score given a higher selection probability. Specifically, given a set of N spanning matches with scores f_1, f_2, \dots, f_N , the probability P that match i will be selected is:

$$P(i) = \frac{f_i}{\sum_{j=1}^N f_j}$$

Cluster Merge Creation

For each potential cluster merge (during fragment or match recombination), we aim to select the best merge by either spanning match or spanning fragment recombination. For each potential cluster, we check if there are equivalent spanning matches in the set of all possible matches that can be inserted. Finding such matches is important because they show whether the new cluster is consistent with existing matches. When the potential merges between a pair of clusters C_i and C_j are compared, we score the merge as $C_{ij}\beta_{ij}$, where C_{ij} is the fitness score of the resulting cluster, and β_{ij} is parameter proportional to the number of inserted equivalent matches, which encourages merges which have a lot of consistent spanning matches to be selected.

4.4. Cluster Optimization and Feasibility

Each potential merge consists of a pair of clusters and a transformation that aligns them, and results in a new child cluster. However, in the large population of resulting child clusters, many are in-feasible, contain global misaligned fragments, or are missing

matches. We thus process the newly created raw population to account for these issues, and generate clusters that result from feasible merges.

Cluster Optimization

Once a cluster is created via a spanning fragment or spanning match merge, it needs to be globally relaxed so that the fragment positions and orientations satisfy the match constraints as well as possible. Each match is a constraint on fragment orientation and translation, and it is difficult to jointly optimize both. The orientation optimization can be decoupled from the translation optimization, and solved first. Given $\theta_1, \theta_2, \dots, \theta_N \in [0, 2\pi]$ unknown fragment orientations (represented by rotation angles), and a set of known (but possibly noisy or incorrect) orientation constraints $\delta_{i,j} = \theta_i - \theta_j$, we can formulate finding optimal orientations as a set of equations:

$$\theta_i - \theta_j = \delta_{i,j} \pmod{2\pi}$$

The above equations contain non-linear constraints that we can linearize them by defining $z_i = e^{i\theta_i}$, and formulating the optimal optimization search as a least squares (LS) objective. LS methods are highly susceptible to outlier effects because the objective function becomes highly dominated by outlier contributions, which results in noisy and incorrect solutions. We employ the robust formulation proposed by [Sin11], which limits the contributions of each incorrect match. The transformed objective is:

$$\max_{\theta_1, \theta_2, \dots, \theta_n \in [0, 2\pi]} \sum_{i,j=1}^n e^{-i\theta_i} H_{ij} e^{i\theta_j} \quad (1)$$

where H is the $n \times n$ complex matrix with entries:

$$H = \begin{cases} e^{i\delta_{ij}}/m(f_i) & \text{if match } \delta_{ij} \text{ is observed} \\ 0 & \text{otherwise} \end{cases}$$

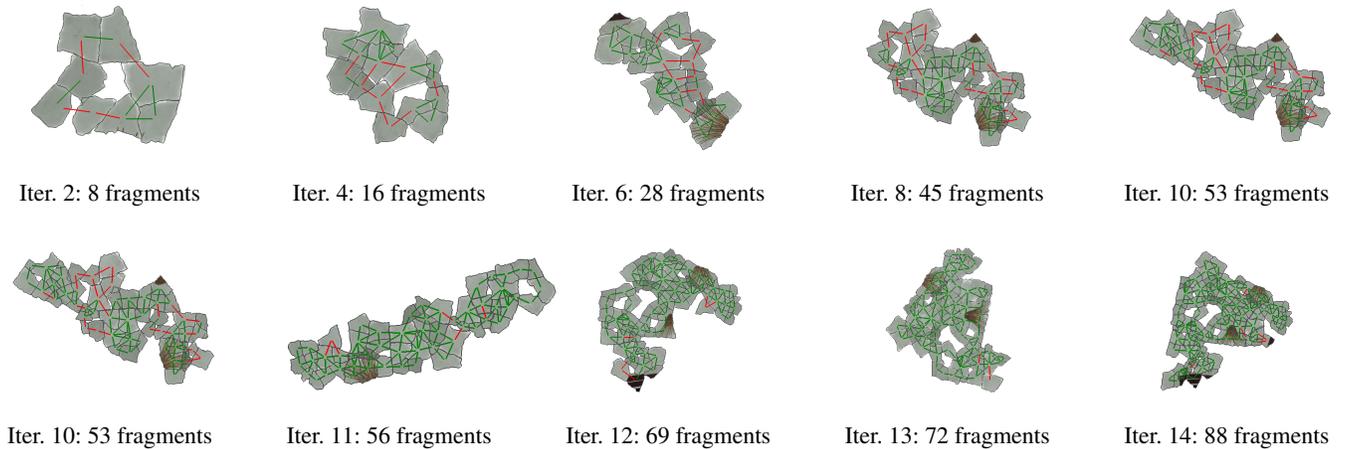


Figure 2: Sample run of our algorithm: evolution of clusters through iterations. Shown above are the top clusters after the end of each respective iteration. The clusters progressively accumulate more correct matches as their consistency is verified with more fragments.

where $m(f_i)$ is the number of matches of fragment f_i (the goal is to equalize the importance of fragments by match degree), which normalizes the contributions of fragments that are connected by different numbers of matches.

We extend this method by formulating an iterative version (Iterative Normalized Eigenvector Method) based on [O*14]. In particular, we iterate the estimation of angles with the subset of match constraints that were respected by the output from the previous iterations i.e. we reject matches that differ from generated angles by more than a threshold T . This process is repeated until the number of rejected matches in each iteration converges to zero. We find that the iterative version further improves the robustness of the method, and generates more accurate solutions.

Feasibility

After the cluster has been created and globally optimized, we check for its feasibility and add it to the population of clusters only if it passes the check. A cluster is considered in-feasible if it would not represent a valid configuration in a real wall painting. We thus prune clusters if they have high overlap to fragment area (where overlap is computed using hexagonal simplifications of fragment areas for fast computation). Clusters that are densely connected via matches and have a high ratio of fragment area to the convex hull are more likely to be correct. Thus we impose constraints on the ratio of fragment covered area to convex hull of the cluster and the maximum allowable number of spanning matches (matches that, if removed, would leave the cluster disconnected). We discuss these parameters in supplementary material.

Fitness Function

An integral part of the evolution is the cluster fitness function, that determines how to rank the clusters with respect to the rest of the generated population.

The population of clusters is sorted according to a fitness function that represents the correctness and size of clusters. Clearly,

correct clusters (i.e. those consisting of mostly correct matches and this fragment placements) need to be assigned high scores so that they are selected in the next round of the selection procedure. The function must also reflect the size of the cluster, as the clusters need to be encouraged to grow in number of fragments, rather to converge to a subset of the fragments. Since clusters that have dense match connections and high scoring matches are typically the best candidate clusters for growth, we incorporate this prior knowledge into the fitness function. Specifically, we calculate the number of fragments $span_{f_i}$ or number of matches $span_{m_i}$ that are part of the spanning tree of cluster C_i , i.e. if removed, the cluster will become disconnected. The goal of the fitness function is to minimize $span_{f_i}$ and $span_{m_i}$, since in clusters that can be easily disconnected, global cluster optimization is unable to detect and remove match error. Let $MaxST(C_i)$ be the sum of the match scores of the maximal spanning tree of cluster C_i . Then the fitness function of the cluster C_i is:

$$f(c_i) = MaxST(C_i) - W(span_{f_i} + span_{m_i})$$

where W is the weighting parameter that controls the importance of the spanning match and fragments.

5. Experimental Results

We conduct experiments to test the proposed method on a challenging benchmark wall painting dataset for which ground truth fragment alignments are available for verification. We compare our reconstruction method to existing techniques used previously for the same dataset. We also quantitatively evaluate the contribution of the individual components of our algorithm (selection, cluster optimization, fitness function). In particular, we evaluate the results using the number of fragments in the largest cluster at convergence as well as the match correctness F-score (harmonic mean of precision and recall of the number of correct matches compared to the total number of matches in the cluster).

All experiments were conducted on Linux servers with 2.6 GHz

Method	# of Fragments	F-score
Our Full System (GA)	90	0.823
Hierarchical Clustering (HC)	42	0.411
Dense Cluster Growth (DCG)	7	0.082

Table 1: Performance comparison of different wall painting reconstruction methods with our fitness and feasibility criteria.

Sandy Bridge processors with 16-core CPU nodes. Selection and recombination procedures were implemented to make use of all parallel threads available.

Dataset

Our experiments are run with data from the Princeton Synthetic Fresco, a plaster wall painting created and fractured by Greek archaeologists for the purpose of testing reconstruction algorithms [BTFN*08]. While much younger than its counterparts (some of which are over 3,000 years old), this wall painting was constructed with a similar process similar to the ones created in ancient times in Akrotiri, Greece. It was artificially weathered to create erosion similar to the originals and thus presents a challenging test case for our system. In all, the wall painting has 129 fragments with 256 manually verified ground-truth matches between adjacent fragments. The fragments are partitioned into three disjoint connected components, of which one is large (114 fragments) and the other two are small – we execute our tests only on the largest connected component, as is performed in [CBR*11].

For each test, we run the fragment matching algorithm of [FSTF*11] to generate a set of 49,153 match proposals, of which we found 180 to be correct. Following the approach of [CBR*11], we augment this set by adding the missing truth matches, so that both methods start from the same initialization for fair comparison.

Comparison to Alternative Methods

The goal of the first experiment is to evaluate the proposed genetic algorithm (GA) approach in comparison to commonly used reconstruction methods: Dense Cluster Growth (DCG) and Hierarchical Clustering (HC).

The Dense Cluster Growth (DCG) [KK01,GMB02,MK03] is a popular algorithm that relies on the notion that given a seed cluster, one can iteratively add the best fragment and then replace the seed cluster with the newly formed cluster. This approach is equivalent to the configuration of our algorithm where only one best cluster is selected at every iteration, and clusters can only be formed by merging with this cluster. This approach is limited by the inability to backtrack after it has added several wrong matches to the seed cluster.

The Hierarchical Clustering (HC) [BK93, ÜT99, PPE*02, BBB05, KK01, MP06, DS09, SE10, CBR*11, SBVV14] is another approach that expands the search space by maintaining a queue of many existing clusters, iteratively merging the best candidate pair of clusters and deleting the two parent clusters. The algorithm converges after no more feasible clusters can be created.

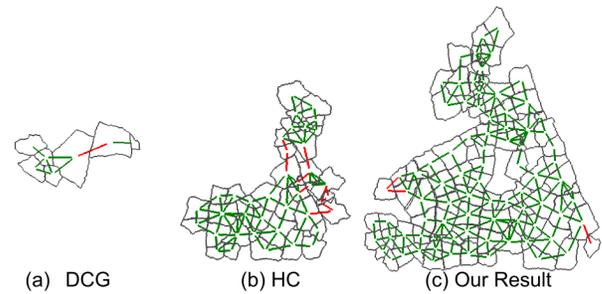


Figure 3: Comparison of reconstruction quality and size of the best runs of Dense Cluster Growth (DCG), Hierarchical Clustering (HC), and our method (GA), corresponding to numerical results in Table 1.

To achieve the most meaningful and fair comparison of the above algorithms with our method, we implemented the above algorithms with the same cluster feasibility conditions and fitness function as used in our algorithm (the choice of these functions will always affect the final reconstruction result for any of the described method). We evaluate the three techniques based on the F-score (harmonic mean of precision and recall) of correct matches in the largest output cluster. Precision is defined to be the proportion of correct matches in the cluster, and recall is the proportion of correct matches retrieved by this cluster. For example, small but mostly correct clusters will be assigned a high precision but low recall score. Because the merging strategy is probabilistic in nature, we chose the best of ten independent trials for our system and the HC algorithm, and the best of all of the 114 available starting points for the DCG algorithm.

A table comparing the number of fragments in largest reconstruction and best F-score for each of the three methods appears in Table 1. The proposed GA approach is able to achieve the highest F-score by a large margin compared to previous approaches, as well as the largest number of fragments in the reconstruction. In Figure 3, we show the corresponding reconstructions. The GA approach is able to achieve the highest combination of precision and recall by a large margin compared to previous approaches. Specifically, our algorithm achieves a reconstruction of 90 fragments, while the HC algorithm reconstructs a subset of 42 fragments and the DCG algorithm reconstructs a subset of 7 fragments.

Comparison to the Previous State-of-the-Art

Next, we compare our reconstruction to the one produced by [CBR*11], the previous best result for the same data set, but using a hierarchical algorithm with a different optimization function and feasibility criteria. Since the code and data for that previous result are not available to us, we can only compare our results visually. Figure 5 shows the best result presented by [CBR*11] in their paper. It can be seen that it contains fewer correctly placed fragments, fewer correct matches, and more misalignment errors than our result. It is interesting to note that the different algorithmic criteria used in the algorithm of [CBR*11] produces a significantly different reconstruction result to our implementation of hierarchical clustering seen in Figure 3 (achieving a larger reconstruction

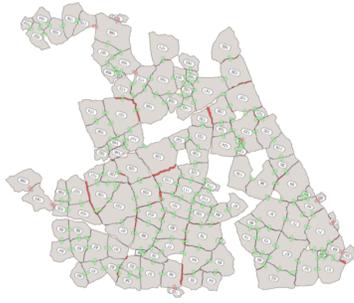


Figure 4: Reconstruction result of [CBR*11]. The final reconstruction contains 118 fragments and 188 matches, ten of which are incorrect. There are subsets of fragments that are incorrectly positioned with respect to each other, as well as clear drift. Image obtained from authors of [CBR*11].

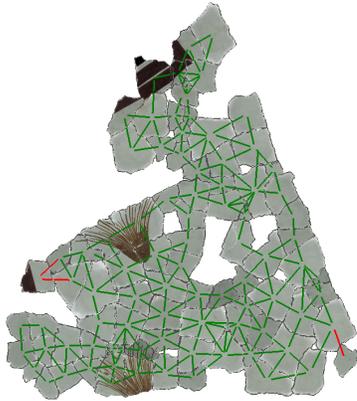


Figure 5: Comparison of Reconstruction methods. Our final reconstruction contains 90 fragments and 187 matches, three of which are incorrect. Wrong matches resulted in only two misplaced fragments that are attached to the boundaries of the wall painting (which in itself is a source of ambiguity).

at a cost of extra error and misalignment), which further highlights the importance of these components to achieving large and correct reconstructions.

Evaluation of Algorithmic Contributions

We evaluate the reconstruction quality of the best reconstruction at each iteration of our algorithm. In Figure 6, we show averaged precision-recall plot for the first 10 iterations of our algorithm, along with error bars that describe the variation of quality (one such longer sample run is shown in Figure 2). The behavior of precision fluctuates at low recall levels (corresponding to plausible but incorrect smaller clusters getting created). However, at larger recall levels, the precision actually increases, which shows that our algorithm is able to verify and build upon the consistency of the best reconstruction. This experiment followed our hypothesis that the fitness of the partial reconstructions, which is difficult to verify using a heuristic evaluation function, can be successfully measured using consistency in a larger reconstruction.

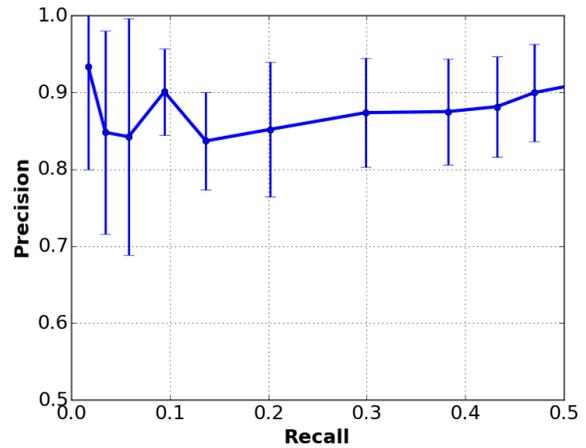


Figure 6: Precision-Recall comparison of different iterations of our algorithm. We aggregate the results from ten independent runs (10 iterations each), and average precision and recall values in each iteration (the vertical bars show the standard deviation).

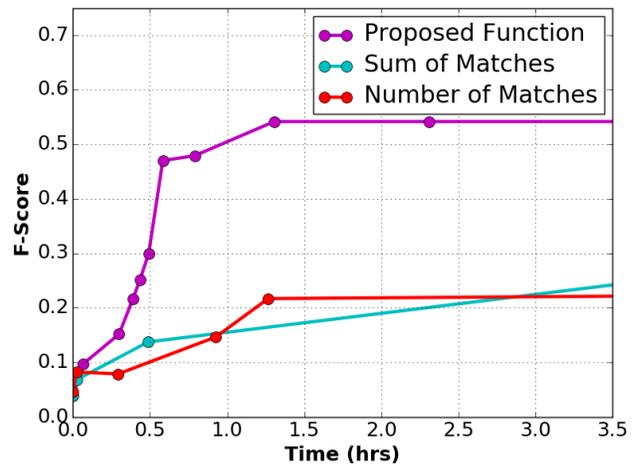


Figure 7: We compare the performance of our method with the proposed fitness function as well as simpler heuristic variants. It can be seen that the fitness function that accounts for connectivity in the clusters significantly outperforms any heuristic functions.

As a final experiment, we evaluate the individual contributions of the algorithmic components of our approach. Specifically, we investigate the effect of the proposed fair selection procedure on reducing the computational time while maintaining the same quality of reconstruction. In Figure 7, we compare the proposed fitness function with simpler variants: (1) sum of match scores in a cluster and (2) the number of fragments. It can be seen that using the proposed fitness function achieves at least a 2x improvement in F-score within the same amount of compute time.

In Figure 8 we compare different optimization approaches used during the recombination step to optimize each new cluster. The

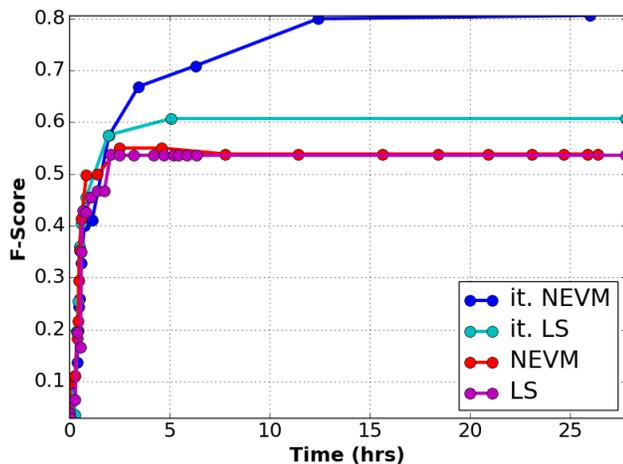


Figure 8: We compare the performance of our method with various cluster optimization methods. The Iterative Normalized Eigenvector Method (it. NEVM) converges at a higher F-score value than any of the other methods, which shows its ability to successfully filter noise and outliers from matches.

proposed Normalized Eigenvector Method (NEVM) achieved a higher F-score than the previously used Least Squares (LS) approach whenever both algorithms were run for the same amount of time. The iterative versions of these algorithms also showed the same trend in performance, however, they were able to continuously increase the size of the correct reconstruction, as seen in Figure 8, when their non-iterative counterparts converged. The proposed Iterative Normalized Eigenvector Method achieved the best reconstruction overall within the time of evaluation.

6. Conclusion and Future Work

We present an approach for wall painting reconstruction using a genetic algorithm. The algorithm recombines clusters of fragments using a novel robust optimization method that iteratively discards outlier matches while optimizing fragment transformations, which allows the algorithm to handle large amounts of noise and to avoid converging into local minima. It addresses the difficulties of an exponentially large search space by favoring the qualities of correctly reconstructed clusters and selecting the clusters with a potential to grow. It also tackles the difficulties of backtracking from plausible but incorrect solutions by maintaining a population of sub-reconstructions at every iteration step. The combination of these methods is an algorithm that can reconstruct more accurate and larger reconstructions than was possible previously.

Our results are far from perfect, and we hope that our findings will stimulate further research into better global assembly techniques for the challenging task of archaeological artifact reconstruction. First, it would be interesting to see how other stochastic optimization techniques can handle and navigate the challenging search space of the problem. Second, we show how robust convex optimization methods can benefit in finding order in challenging and noisy pairwise match sets, and it would be interesting to see

how other such techniques could be adapted for this task. Finally, given the importance of the fitness and feasibility criteria for partial reconstructions, it would be interesting to learn these functions from existing artifacts that contain ground truth and apply them to unsolved wall paintings.

7. Acknowledgements

We wish to thank Professor Christos Doumas, Andreas Vlachopoulos, and the conservators and archaeologists at the Akrotiri Excavation Laboratory of Wall Paintings for their input and collaboration. We thank Amit Singer, Onur Ozyesil, Antonio García Castañeda, Tim Weyrich, Szymon Rusinkiewicz, Benedict Brown, and the Princeton Graphics group for research discussions. Finally, we thank Dimitris Gondicas, Peter Nomikos Jr., The Kress Foundation, Seeger Foundation, Thera Foundation, Cotsen Family Foundation, and National Science Foundation (NSF-GRFP and IIS-1251217) for their support.

References

- [BBB05] BISWAS A., BHOWMICK P., BHATTACHARYA B.: Reconstruction of torn documents using contour maps. In *IEEE International Conference on Image Processing (ICIP)* (2005). 1, 2, 6
- [BK93] BUNKE H., KAUFMANN G.: Jigsaw puzzle solving using approximate string matching and best-first search. In *Computer Analysis of Images and Patterns* (1993). 2, 6
- [BLD*12] BROWN B., LAKEN L., DUTRÉ P., VAN GOOL L., RUSINKIEWICZ S., WEYRICH T.: Tools for virtual reassembly of fresco fragments. *International journal of heritage in the digital era* (2012). 1
- [BTFN*08] BROWN B. J., TOLER-FRANKLIN C., NEHAB D., BURNS M., DOBKIN D., VLACHOPOULOS A., DOUMAS C., RUSINKIEWICZ S., WEYRICH T.: A system for high-volume acquisition and matching of fresco fragments: Reassembling Thera wall paintings. In *ACM Transactions on Graphics (TOG)* (2008). 2, 6
- [CBR*11] CASTAÑEDA A. G., BROWN B. J., RUSINKIEWICZ S., FUNKHOUSER T. A., WEYRICH T.: Global consistency in the automatic assembly of fragmented artefacts. In *International conference on Virtual Reality, Archaeology and Cultural Heritage (VAST)* (2011). 2, 3, 6, 7
- [CWA*01] COOPER D. B., WILLIS A., ANDREWS S., BAKER J., CAO Y., HAN D., KANG K., KONG W., LEYMARIE F. F., ORRIOLS X., VELIPASALAR S., VOTE E. L., JOUKOWSKY M. S., KIMIA B. B., LAIDLAW D. H., MUMFORD D.: Assembling virtual pots from 3D measurements of their fragments. In *Conference on Virtual Reality, Archeology, and Cultural Heritage* (2001). 2
- [DGLS02] DA GAMA LEITAO H. C., STOLFI J.: A multiscale method for the reassembly of two-dimensional fragmented objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2002). 2
- [Dou92] DOUMAS C.: *The Wall-Paintings of Thera*. Thera Foundation, 1992. 1
- [DS09] DE SMET P.: Semi-automatic forensic reconstruction of ripped-up documents. In *International Conference on Document Analysis and Recognition (ICDAR)* (2009). 2, 6
- [FG64] FREEMAN H., GARDER L.: Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition. *IEEE Transactions on Electronic Computers* (1964). 2
- [FSTF*11] FUNKHOUSER T., SHIN H., TOLER-FRANKLIN C., CASTAÑEDA A. G., BROWN B., DOBKIN D., RUSINKIEWICZ S., WEYRICH T.: Learning how to match fresco fragments. *Journal on Computing and Cultural Heritage (JOCC)* (2011). 2, 3, 6

- [FT05] FORNASIER M., TONIOLO D.: Fast, robust and efficient 2D pattern recognition for re-assembling fragmented images. *Pattern Recognition* (2005). 2
- [GMB02] GOLDBERG D., MALON C., BERN M.: A global approach to automatic solution of jigsaw puzzles. In *Annual Symposium on Computational Geometry* (2002). 2, 6
- [HFG*06] HUANG Q.-X., FLÖRY S., GELFAND N., HOFER M., POTTMANN H.: Reassembling fractured objects by geometric matching. *ACM Transactions on Graphics (TOG)* (2006). 2
- [HKH*10] HENRY P., KRAININ M., HERBST E., REN X., FOX D.: Rgb-d mapping: Using depth cameras for dense 3D modeling of indoor environments. In *International Symposium on Experimental Robotics (ISER)* (2010). 1
- [Joh92] JOHN H.: *Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, 1992. 2, 3
- [KK01] KONG W., KIMIA B. B.: On solving 2D and 3D puzzles using curve matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (2001). 2, 6
- [KL06] KOLLER D., LEVOY M.: Computer-aided reconstruction and new matches in the Forma Urbis Romae. *Bullettino Della Commissione Archeologica Comunale di Roma* (2006). 1
- [KS09] KLEBER F., SABLATNIG R.: A survey of techniques for document and archaeology artefact reconstruction. In *International Conference on Document Analysis and Recognition (ICDAR)* (2009). 2
- [LGJ*07] LU Y., GARDNER H., JIN H., LIU N., HAWKINS R., FARRINGTON I.: Interactive reconstruction of archaeological fragments in a collaborative environment. In *Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications* (2007). 2
- [LK*81] LUCAS B. D., KANADE T., ET AL.: An iterative image registration technique with an application to stereo vision. In *IJCAI* (1981). 1
- [MK03] MCBRIDE J. C., KIMIA B. B.: Archaeological fragment reconstruction using curve-matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) Workshop* (2003). 2, 6
- [MP06] MAKRIDIS M., PAPAMARKOS N.: A new technique for solving a jigsaw puzzle. In *IEEE International Conference on Image Processing (ICIP)* (2006). 2, 6
- [O*14] OZYESIL O., ET AL.: Camera motion estimation by convex programming. 5
- [PK03] PAPAIOANNOU G., KARABASSI E.-A.: On the automatic assembly of arbitrary broken solid artefacts. *Image and Vision Computing* (2003). 2
- [PPE*02] PAPAODYSEUS C., PANAGOPOULOS T., EXARHOS M., TRIANTAFILLOU C., FRAGOULIS D., DOUMAS C.: Contour-shape based reconstruction of fragmented, 1600 bc wall paintings. *IEEE Transactions on Signal Processing* (2002). 2, 6
- [RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the ICP algorithm. In *International Conference on 3-D Digital Imaging and Modeling* (2001). 1
- [SBVV14] SANCHEZ-BELENGUER C., VENDRELL-VIDAL E.: An efficient technique to recompose archaeological artifacts from fragments. In *International Conference on Virtual Systems & Multimedia (VSM)* (2014). 2, 6
- [SDN13] SHOLOMON D., DAVID O., NETANYAHU N. S.: A genetic algorithm-based solver for very large jigsaw puzzles. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (2013). 1, 3
- [SE06] SAĞIROĞLU M. Ş., ERÇİL A.: A texture based matching approach for automated assembly of puzzles. In *International Conference on Pattern Recognition (ICPR)* (2006). 2
- [SE10] SAĞIROĞLU M. Ş., ERÇİL A.: Optimization for automated assembly of puzzles. *Top* (2010). 2, 6
- [Sin11] SINGER A.: Angular synchronization by eigenvectors and semidefinite programming. *Applied and Computational Harmonic Analysis* (2011). 4
- [TFBW*10] TOLER-FRANKLIN C., BROWN B., WEYRICH T., FUNKHOUSER T., RUSINKIEWICZ S.: Multi-feature matching of fresco fragments. *ACM Transactions on Graphics (TOG)* (2010). 2
- [TFSM02] TOYAMA F., FUJIKI Y., SHOJI K., MIYAMICHI J.: Assembly of puzzles using a genetic algorithm. In *International Conference on Pattern Recognition* (2002). 3
- [URD*04] UKOVICH A., RAMPONI G., DOULAVERAKIS H., KOMPATSIARIS Y., STRINTZIS M.: Shredded document reconstruction using MPEG-7 standard descriptors. In *IEEE International Symposium on Signal Processing and Information Technology* (2004). 2
- [ÜT99] ÜÇOLUK G., TOROSLU I. H.: Automatic reconstruction of broken 3-D surface objects. *Computers & Graphics* (1999). 2, 6
- [VB12] VIDAL E. V., BELENGUER C. S.: Automatic archaeological fragment reconstruction using hierarchical depth maps. In *Conferencia Latinoamericana En Informatica (CLEI)* (2012). 2
- [VVS14] VENDRELL-VIDAL E., SÁNCHEZ-BELENGUER C.: A discrete approach for pairwise matching of archaeological fragments. *Journal on Computing and Cultural Heritage (JOCCH)* (2014). 2
- [WC08] WILLIS A., COOPER D.: Computational reconstruction of ancient artifacts. *IEEE Signal Processing Magazine* (2008). 2
- [WSKL88] WOLFSON H., SCHONBERG E., KALVIN A., LAMDAN Y.: Solving jigsaw puzzles by computer. *Annals of Operations Research* (1988). 2