A computer-assisted constraint-based system for assembling fragmented objects

Gregorio Palmas[1,2], Nico Pietroni[1], Paolo Cignoni[1], Roberto Scopigno[1]
[1] Visual Computing Lab, CNR-ISTI, Pisa, Italy
[2] Max-Plank-Institute fur Informatik, Saarbruecken, Germany
Email: firstname.familyname@isti.cnr.it

***Abstract*—We propose a computer-assisted constraint-based methodology for virtual reassembly of Cultural Heritage (CH) artworks. Instead than focusing on automatic, unassisted re-assembly, we targeted the scenarios where the reconstruction process is not be based on shape properties only but it is build over the experience and intuition of a CH expert. Our purpose is therefore to design a flexible interactive system, based on the selection of a set of constraints which relates different fragments, according to the understanding and experience of the CH operator. Once the user has defined those constraints, the system searches for a suitable solution, using a global energy minimization strategy that considers simultaneously all the pieces involved in the reconstruction process. Additionally, our framework provides the possibility to work in a hierarchical way, mimicking the traditional physical procedure that archaeologists use to reassemble tangible fractured objects.**

The frameworks is designed to work even with fragments that could have been severely damaged or eroded. On those datasets, automatic approaches may often fail, since the fractured regions do not contain enough geometric information to infer the correct matches.

We present some successful uses of our framework on real application scenarios.

I. Introduction

The progress of the technologies for the acquisition of high-quality 3D models of real CH artworks has been impressive. The adoption of active or passive approaches for sampling the shape of artworks is becoming a common task in several CH activities, such as: study/research, documentation of findings in excavations, virtual presentation and restoration. 3D digital models provides several important advantages, such as the independence from time and space constraints. Scholars may exploit a wide and common knowledge base, due to the availability of enhanced searching engine over digital repositories of CH assets, interactive visual analysis even on the web, flexible tools for shape comparisons and improved shape reasoning capabilities. All these technologies are going to become available and accessible. One interesting application of visual technology to CH is the virtual reassembly of fragmented artworks.

A. Motivation

Reconstructing fragmented objects is a common problem for archaeologists or restorers. Ancient artifacts are often discovered in a fragmented status, broken apart by human or natural intervention; those fragments are inevitably eroded and/or incomplete. The reconstruction process performed by archaeologists or restores consists in four stages:

1) Visual fragment analysis;
2) Devising matching hypotheses;
3) Rehearsing and cross-comparing this hypotheses on the real fragments (an action that can introduce further degradation);
4) Reunion of adjoining pieces according to the assessed reconstruction hypothesis.

Each reconstruction phase presents several issues which are strictly related to the difficulty of the manual work, indeed the matching process is executed by manually checking each candidate matching pair. Moreover, most of the non-trivial cases require the design of a supporting structure to dispose the fragments according to a temporary global recombination layout and to support restorer's assessment.

Being the classical approach very labour-intensive, this task was the focus of several research initiative, with the aim of designing computer-aided methods which should overcome some of the practical problems related to the manual reconstruction. Most of these approaches are focusing on the automation of the process, whie a beware focusing on the design of specific GUIs or haptic systems.

B. Contribution

We propose an assisted virtual reconstruction method, focusing our attention on the inclusion of the user experience and intuition in the processing loop. The majority of the current computer-assisted methods proposes automatic computation of a complete/partial reassembly solution, which is determined uniquely by geometric characteristics. Conversely, our approach provides the user an interactive system (see Figure 1) with enough degree of freedom to use his experience, by suggesting possible pair-wise connections or more generic constraints between fragments. The idea is not to oblige the user to align each pair of fragments in the correct matching position, but to offer a flexible and extendable set of operators that allows to express constraints to be used to find a plausible respective position of the fragments. In our vision, several constraints can be specified and offered to the operator to define the scenario that should lead to the computation of a possible solution (see Section III). In the current prototypal system we have incorporated only the *adjacency* constraint, that allows the user to specify pairs of expected adjoining areas between two fragments, see Figure 2. These selection of possibly correspondent areas is used by the system to compute the rigid transformation for each connected entity that allows to support an optimal mapping over the set of constraints given in input by the user. Our method is designed to be user-assisted,

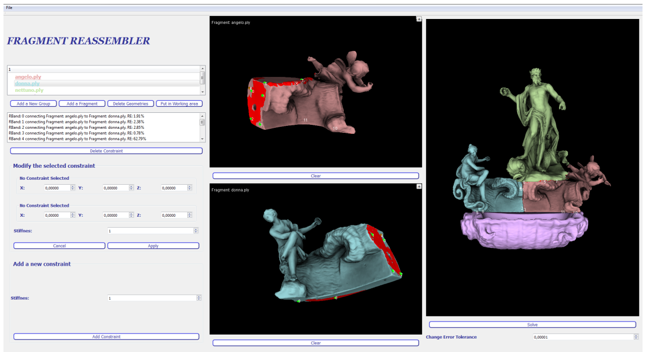

Fig. 1. A snapshot of our Fragment Reassembler tool

however is possible to extend it to include automatic feature matching strategies, such as [6]. Another important aspect of our design is to support a hierarchical approach, taking inspiration from the classical restoration methodologies. The user may organize the work by subsets, treating subgroups in an independent manner and then assembling those groups using the same operators used to solve pair-wise recombination. Once a group has been recombined, it can be considered by the system as a single composed fragment. This feature implies that the assembling procedure can be modeled as a tree which has fragments at the leaves, groups as intermediate nodes and the reconstructed object as the root.

The system provides also some tools to verify the accuracy of the solution in terms of interpenetration detection and residual distance visualization.

We verified the usability of our framework on several real application scenarios. Among them, we report here results from: the restoration of the Madonna of Pietranico (L'Aquila, Italy), a 15th century terracotta statue, which was severely damaged during the 2009 earthquake[1]; the study of a complex object part of the Victoria and Albert Museum collection (London, UK).

II. State of The Art

In the following subsections we present a brief description of the state of the art, which is also the subject of a survey paper [2].

A. The Classical Physical Approach

With the term "classical approach" we refer to the methods currently used by restorers or archaeologist, focused on physical reconstruction methods which do not involve the use of digital models. This class of methods involves a direct manual work and the use of consolidated technologies, such as photography. Usually the reconstruction process starts directly at the field excavation, where all the discovered fragments are meticulously classified. In such stage the archaeologists establish categorization and chronological sequences of vessel types that have been developed during the occupation timeline of a excavation field [3]. This cataloging process is performed by examining a few precise features of indicative fragments. After this cataloguing phase, the archaeologists proceed by physically matching pairs of fragments, choosing them among the set of possible candidates.

The main issues in classical approaches are: the experience expected from the operator, the time needed, the potential degradation introduced (in the case of very fragile fragments), the complexity of manipulation (e.g. for heavy fragments), and the need of building supporting structures (a complex and time-consuming sub-task).

Fig. 3. An image presented in [3] to show an intermediate reconstruction stage, with the set of sherds and some partially reconstructed regions of a vessel.

B. Automatic Methods

Automatic methods aim to compute the reassembled position of each fragment by identifying corresponding geometric or appearance features between different fragments. Hence, the crucial step is to provide a robust, yet expressively powerful shape descriptor to identify matching features coming from different fractured regions. Shape descriptors are usually based on a local characterization of geometric neighborhoods which should be invariant with respect to rigid transformations. Under these assumptions, the problem is somehow similar to the automatic registration of range scan [4], [5].

Huang et al. [6] presented a reconstruction pipeline composed by a segmentation algorithm (used to identifying fracture surfaces), followed by a feature-based robust global registration for pairwise matching of fragments, and by a simultaneous local registration of multiple fragments. Toler et al. [7] proposed a multiple-feature approach for determining matches between small fragments of archaeological artifacts such as frescoes ; they introduce a set of feature descriptors that are based not only on color and shape, but take into account also normal maps. Another study on computer-assisted reassembly of frescoes fragments has been presented in [8]. A similar approach has been proposed in [9] for the reconstruction of the Severan Marble Plan of Rome. McBride et al [10] proposed a method based on two stages: they first compare every pair of fragments and use partial curve matching to find similar portions of their respective boundaries; in the second stage they search for a globally optimal arrangement which is based on a best-first strategy to attach fragments with the highest pairwise affinity. Winkelbach et al [11] proposed an automatic method based on matching of clusters of points organized as a hierarchical structure. Belenguer et al. [12] proposed an automatic matching approach using a shape-descriptor based on a discrete sampling of the fracture surface.

The above methods rely on the capability of finding a good match of geometric/appearance characteristics over the fragments. However, in many case missing portions and eroded surfaces make this assumption rather unpractical.

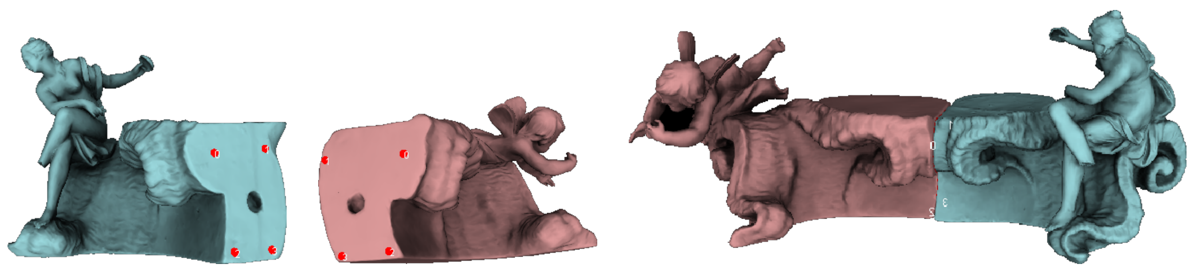

Fig. 2. The user specifies *adjacency* constraints by setting pairs of points that are supposed to be *close* each other (and not coincident) in the final configuration.

C. Semi-Assisted methods

Semi-assisted methods put the user into the reconstruction loop, i.e. using his experience to drive the system towards a plausible solution. The user could suggest to the system possible matches or some constraints to influence the reconstruction process. Involving the user in the reconstruction loop is mandatory for cases where missing or highly damaged pieces are part of the puzzle. Moreover, user's experience can even help to improve the efficiency of the reconstruction process. If we introduce constraints which depend on user's experience and knowledge, the search for a plausible solution could be much faster than for a pure automatic system.

Papaioannou et al [13] presented a semi-automatic reassembly procedure based on artificial intelligence algorithms working on geometrical information. Parikh et al [14] proposed an approach where the user can easily assemble a desired object from a large collection of pieces (many of which are irrelevant) by iteratively selecting compatible parts. Mellado et al [15] proposed a method based on a real-time interaction and manipulation loop: an expert user steadily specifies approximate initial relative positions and orientations between two fragments by means of a tangible user interface. These initial poses are continuously improved and validated in real-time by the system.

III. Our scenario

Our reconstruction scenario is based on an interactive approach. Our system should support the user in the specification of the rules and actions which, according on his experience, should guide the reassembly of the fragmented artwork.
The entities which are involved in our process are:

- **Pieces**: digitized 3D fragments, encoded with triangle meshes.
- **Sample**: a point placed on the surface of a piece. The list of samples belonging to the fragment A is denoted as $s_0^A ... s_n^A$.
- **Constraint**: a rule that defines the spatial relation between two fragments; a single constraint can completely or even only partially specify the way in which two fragments are related.
- **Groups**: a set of fragments interconnected by constraints can be grouped to form a solid entity. A group may be also composed by other groups. Groups are fundamental for hierarchical fragments reassembly.

In our scenario, the list of constraints that should be offered to the user might be wide; the selection of a minimal subset is a critical action, since extending the set will increase the flexibility and descriptive power of the system, but at the same time increasing the complexity of implementation. We list here some possible constraints:

- **Adjacency**: the user defines a couple of samples, belonging to different fragments, possibly located on corresponding fracture surfaces; the goal of this constraint is to keep the selected pair of samples as close as possible while computing the optimal spatial configuration of the respective fragments. The *Adjacency Costraint* could be enriched by a weight value, expressing the importance of this single constraint when we find a solution to a multi-constraints problem.
- **Area-to-area**: similar to the *Adjacency* constraint, this is a constraint which creates a link between corresponding surface parcels defined over two fragments; in this case the user selects a surface portion on each piece rather than a single sample point (again, usually on the fractured surface). This constrain could be used in the automatic optimization process by searching the best matching features in the corresponding regions and trying to detect an optimal adjoining pose.
- **Surface iso-planarity**: the user selects two corresponding regions on two pieces and states that these two regions should have similar average direction of the normal vector. Introducing this constrain allows to force to displace those pieces in 3D space only on configurations that will preserve the co-planarity over those selected regions. Co-planarity could be verified as an hard constrain (the two regions should lie on the same plane) or a weaker one (they could be on parallel, iso-oriented planes). This constrain could allow the user to define the respective orientation of two pieces even in presence of a highly degraded fractured surface or in the case of a missing connecting fragment.
- **Distance**: the user specifies two fragments and two location over this pieces, and then defines the ideal distance between this fragments in the overall recombination. The goal of this constraint is to allow the user to specify how to compose fragments in the case in which we have missing elements, allowing to define the bridge that allows to relate components which are not connected by available fragments (e.g. a broken hand to be reconnected to a statue even if we do not have anymore a portion of the broken arm). It is ideal

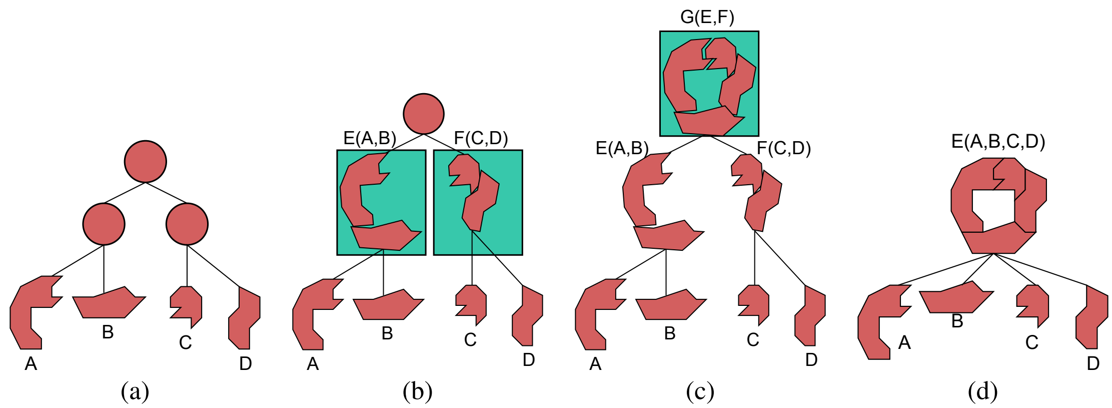

Fig. 4. A simple 2D reconstruction scenario: (a) The constraints graph express the hierarchy of pieces and constraints; (b) according to such hierarchical structure fragment A is matched with fragment B, and similarly fragment C with D; (c) The system propagates the solution to the upper levels of the hierarchy, solving for the groups; (d) The hierarchy is deleted and constraints are minimized all at once, by producing a more globally consistent solution.

if used in conjunction with the *Surface iso-planarity* constraint.

Each of these constraints require a specific user interface component for selecting its parameters. All user-selected constraints can be encoded with a graph.

In our prototypal implementation we have included so far only the *Adjacency constraints*. The extension of our system to other constraints is part of our future work. In the following thus we describe how do we represent the Constrain Graph and how do we solve the recombination problem in the case of *Adjacency constraints*. It is important to state explicitly that the only interactive part is the definition of the constraints that interlink the fragments. All other steps (construction of the graph, solution of the problem by energy minimization) are completely automatic.

A. Constraints Graph

The *constraints graph* encodes the relation interlinking pieces, groups and constraints. This graph encodes the hierarchy between different fragments or groups of fragments, with the latter interconnected by constraints. The constraints graph has one node for each fragment (or group) and one edge for each constraint. The constraints graph defines the *solving sequence*, i.e. the sequence of constraints that should be satisfied to find a proper reassembly configuration, according to the user-defined hierarchy.
It should be noted that, in order to preserve the uniqueness of the solving sequence, this graph must necessarily be a tree; this condition guarantees the uniqueness and the coherence of the solving sequence. In our scenario, to satisfy a specific Adjacency constraint means to retrieve the rigid transformation that minimizes the distance between its samples.
We refer to the example of Figure 4 to explain the concept of a solving sequence:

1) In the first step, according to a hierarchical strategy, we compute and apply the transformations relative to pairs of fragments at the lower levels, i.e. the leaves of the Constraints Graph. Therefore, we retrieve the transformations that satisfy constraints between fragments A and B, and fragments C and D (Figure 4.b).
2) The system repeats the previous step by considering the groups formed by fragments that have been solved in the previous step. A solution is found to reassemble group E and F. Note that, when a transformation is applied to a group, it is implicitly inherited by all of its members. The example in Figure 4.c shows how the group formed by the fragment previously assembled could be matched with another one.
3) Those two steps are repeated until all the involved fragments are placed in their final position (according to the structure of the Constraints Graph specified by the user) and the object is reassembled.
4) Finally, the user can delete the group hierarchy and minimize again the global energy. The resulting transformations will provide a more globally coherent solution (see Figure 4.d).

B. Energy minimization

For each adjacency constraint, we minimize the distance between its two samples. In this sense, we associate to each constraint a local energy term, defined as the squared distance between its samples (multiplied by its weight factor). From a global perspective, we minimize the sum of all per-constraint local energy contributes. For the sake of simplicity, we consider in the following a setup composed by only two fragments, A and B, and n constraints. The energy term can be formulated as:

$$E = \sum (s_i^A - s_{i'}^B)^2 \cdot k_{(i,i')} \forall constraints(s_i^A, s_{i'}^B) \in [1, n] \quad (1)$$

where $k_{(i,i')}$ is the stiffness factor of the constraint $(s_i^A, s_{i'}^B)$. This formulation can be extended to an arbitrary number of pieces.

C. Dealing with rigid transformations

We associate a rigid transformation, i.e. a transformation which does not have any scaling or skewing factor, to each fragment or group (which represents a rigid entity). For a given fragment (or group) A, the rigid transformation is therefore defined as a 3×3 rotation matrix R^A and a translation vector T^A. We can consequently reformulate the energy term of Equation 1 as follows:

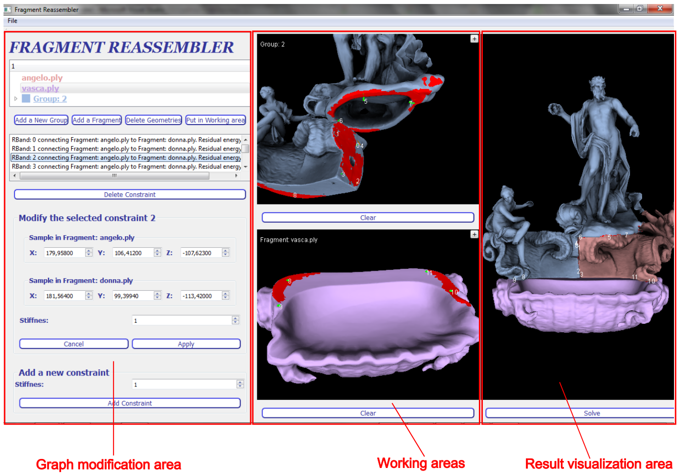

Fig. 5. A screenshot of the user interface of our Fragment Reassembler tool.

$$E = \sum (R^A \cdot s_i^A + T^A - R^B \cdot s_{i'}^B + T^B)^2 \cdot k_{(i,i')} \quad (2)$$

It is important to note that, since the rigid transformation (R, T) is unique for each fragment, this constraints all the samples belonging to a given fragment to move rigidly.
This problems is solved by defining a linear system that includes either the least squares energy term of Equation 1, and the linear constraints that relates the each rigid transformations to corresponding samples.
More precisely, we introduce a linear constraint for each sample s_i^A :

$$R^A \cdot s_i^A + T^A = s_i'^A \quad (3)$$

Then we minimize energy term of Equation 1 by considering all the samples $s_i'^A$ for whom the rigid transformation has been applied. Finally, to make the system solvable, it is sufficient to fix a fragment on a default position, and force its rigid transformation to be the identity.
At this point, the output of our minimization process is a set of transformations (R^A, T^A), one for each piece A. However, we do not ensure the obtained transformations to be rigid. As the properties of rotation matrices $det(R) = 1$ and $R_q^T = R^{-1}$ are not linear, then they cannot be imposed directly as a linear constraint: thus we have no guarantee the obtained transformation to be rigid. Then we use an iterative method to find the rotation matrix R'^A which is the most similar to the generic 3×3 matrix R^A. Once we found the rotation matrix, we update the samples and iterate the process until convergence. Convergence is achieved by checking if the residual energy due to all the constraints is increased during the optimization process.
The system can be further optimized by encoding rotations by relying on skew symmetric matrices. This makes the number of variables encoding each rotation matrix to be reduced from 9 to 3. Skew symmetric matrices linearly approximate infinitive rotations.
In order to solve the overall system we used the Eigen[**?**] linear algebra C++ library. Since the overall system is symmetric positive-definite, we used a Cholesky decomposition.

D. Comparing with range map alignment approaches

Since we have implemented so far only the Adjacency constraint, there could be a number of affinities between our approach and the canonical problem of aligning a few range maps, therefore some clarification is needed. Range maps alignment solutions are based first on a pair-wise iteration of the ICP algorithm [16], that can find a good pairwise transformation that brings two similar maps together, and then on a global registration step to distribute the error introduced in the pairwise matching phase [17].
In our case we start from a very different premise: we cannot rely on the geometric properties of the surface but only on the matching points indicated by a skilled users, so the ICP approach is of little use (adjoining surfaces can be degraded and eroded). Similarly, in the second global phase we proceed in a more structured way, by defining the problem as a simultaneous minimization problem that can take into account the weight/importance of the constraints as indicated by the user.

IV. THE USER INTERFACE OF OUR PROTOTYPE TOOL

Given our reassembly approach, the design and implementation of the user interface of the interactive tool assumes a crucial role. The user must be able to manipulate fragments and to assign efficiently the required constraints. The interface we implemented supports real time creation or groups, which connects fragments by using constraints, exploiting the hierarchical nature of the recombination process. In addition, after the system derived the final position of each fragment (via energy minimization, as explained in Section III), the tool offers instruments to check and quantify the quality of the current reassembly results.

The interface is divided in tree macro-areas (see Figure 5):

- *Graph modification Area.* This area is designed to define the constraint graph. It allows to: create a new group or to split an existing one; add, delete o modify an existing constraint.
- *Working Areas.* This area allows the user to select points on two different fragments or groups in order to define new adjacency constraints. Those two areas act like the hands of the restorer. Their main purpose is to allow him to explore each piece of the current pair individually and pick the samples from different pieces that he/she guesses to be close each other.
- *Result Visualization Area.* This area visualizes the intermediate and final result of the reconstruction process (i.e. the results of each energy minimization iteration).

A. Results validation

The validation of the final results requires specific attention and, hopefully, should be also assisted by the system. Since recombination is a user-driven interactive approach, the user may introduces inconsistencies between constraints. We provide two main tools to visualize such inconsistencies:

- *Residual Energy Visualization.* Constraints are visualized as colored edges that connects spherical dots (samples) in the result visualization area. We associated a color proportional to the residual energy of the constraint after the minimization process (blue is low energy, while red means hight energy).

- *Interpenetration.* It may happen that inconsistent constraints could create interpenetration between fragments. These cases should be detected and corrected. We detect such situations by using a depth-peeling approach implemented on GPU. The interpenetrations are colored in red in the 3D contexts of our interface (Constraints Insertion Area and Result Visualization Area). Figure 6 shows an interpenetration example.

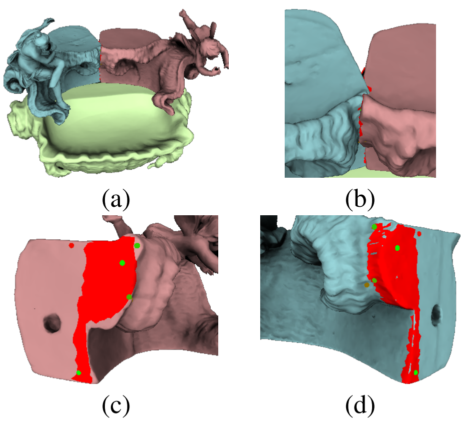

Fig. 6. An example of an incorrect solution where two fragments are interpenetrating each other: (a) global view of the recombination of three fragments; (b) closeup of the interpenetrating region; (c) and (d) the interpenetrating regions of each piece are highlighted in red.

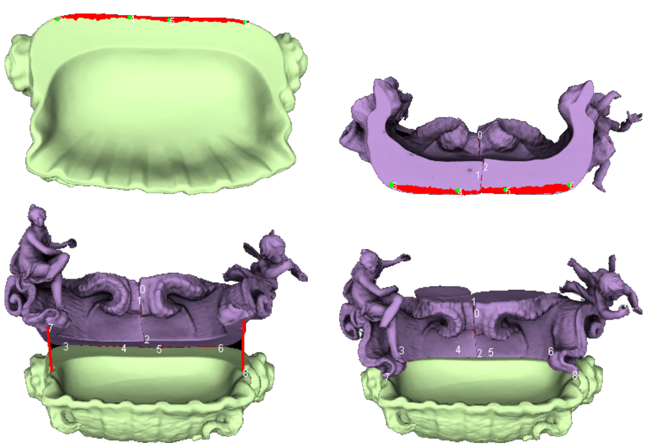

Fig. 7. If the constraints are not uniformly distributed on the fragments, then it may happens that the pieces will no match precisely in their final position (see the red regions in the two top-most images). In this case, it is sufficient to provides a few additional constraints (see the two pairs of adjacency constraints visualized with the red lines in the left-bottom image) to force the pieces to be attached each other correctly (right-bottom image).

Instantiating an insufficient number of constraints, or defining incorrect locations of the corresponding points, may produce a poor layout of the fragments. The example in Figure 7 shows that a wrong placement of constraints may result in a misalignment of some pieces. In this case, the user can fix the problem by adding just two additional constraints (see the red lines in Figure 7).

B. *Saving and loading of intermediate states*

As the reconstruction process can be very long and complicated, any intermediate state of the process can be saved and restored. The possibility of saving and loading a project has been provided by generating an XML file which contains all the needed information to restore the current work (the fragments loaded, the hierarchy of the reconstruction graph and the points on which the constraints have been defined). The user can also export the final work as a MeshLab[18] project, to use MeshLab either for visualization of results or for applying further geometric processing filters.

V. RESULTS

We tested our framework on several real cases. We would like to emphasize that two out of the three examples we show here are real application scenarios that have been commissioned us by restorers or curators, while the last example is a laboratory test. For all the test cases, our system is able to reassemble the final object with a few user interaction gestures.

Fig. 9. Pietranico Madonna, a terracotta statue severely damaged during the 2009 earthquake in Abruzzo, Italy.

The first case study is the *Madonna of Pietranico*. This artwork is a devotional terracotta statue of the 15^{th} century, originally located in the main church of the Pietranico village in Abruzzo, Italy. This statue was severely damaged during the 2009 earthquake. It was fragmented in many pieces of different size (see Figure 9). Those pieces are very eroded (because, being an old terracotta, the fractures produced many small material chips and thus the adjoining surfaces have missing material in-between). Moreover, several small-size fragments are missing. Deriving all the matches between pieces is therefore a complex task for a purely automatic approach. Conversely, our user-assisted approach performed nicely. Notice that the upper and lower parts of the sculpture were processed separately, since they were originally produced as two independent portions. The upper part of this sculpture is composed by 12 pieces, while the lower part required to reassemble 5 pieces.
Our user needed about 3 minutes to complete the reassembly of the lower part and about 20 minutes for the upper part. We provide some statistics on the minimization process in Table I. The performances of the solver are very good: even in the more complex case, like the upper part of the Madonna (12 pieces which required the definition of 49 constraints) the minimization process converges, after few iterations, in just 0.5 seconds.

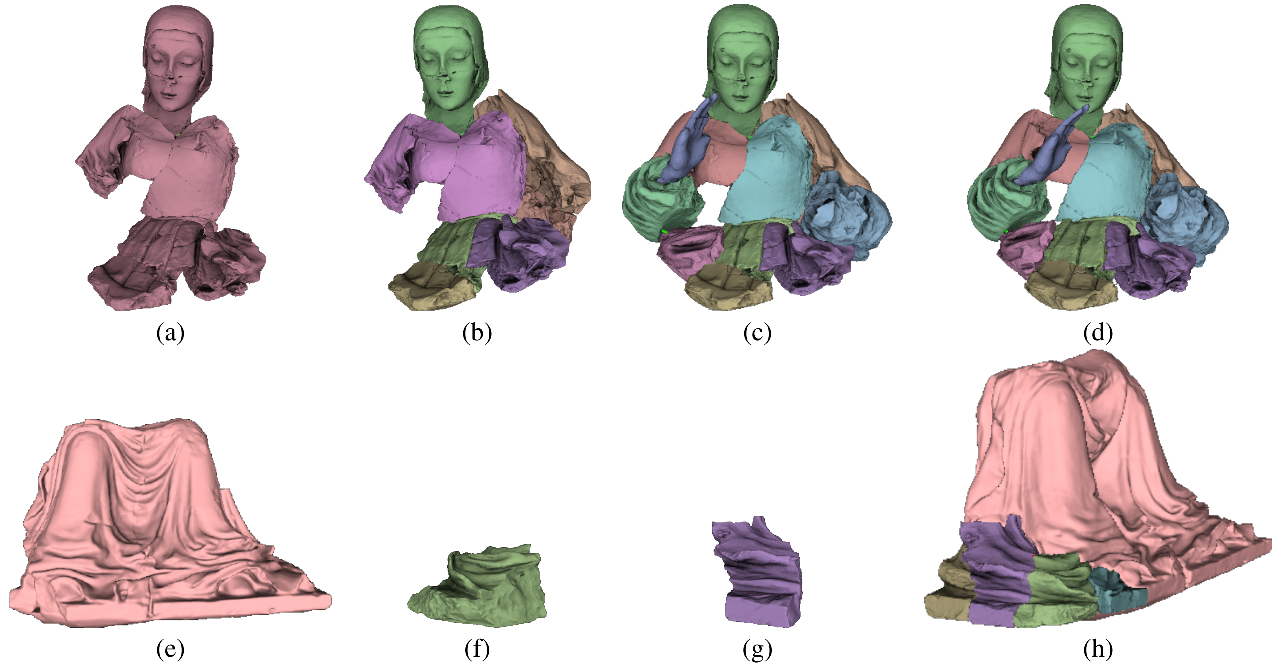

Fig. 8. Reassembly of the upper section of the Pietranico Madonna (a..d): intermediate results in (a), (b) and (c); image (d) shows the result after the global minimization step, tapplied to the intermediate result presented in (c). Image (h) presents the reassembly of the lower part, from several fragments (images e, f and g).

Ph.	# Entities	# Con	Time (sec)	# It.
1	2	4	0.021	5 %
2	2	4	0.01	3 %
3	2	4	0.013	3 %
4	2	4	0.009	2 %
5	5	16	0.16	3 %

TABLE I. DETAILS ON THE RECONSTRUCTION OF THE LOWER PART OF THE MADONNA OF PIETRANICO: THE TABLE REPORTS FOR EACH PHASE OF THE PROCESS (PH) THE NUMBER OF ENTITIES INVOLVED (# ENTITIES), THE NUMBER OF CONSTRAINTS (# CON), THE TIME NEEDED BY THE SOLVER TO PROVIDE A SOLUTION (TIME) AND FINALLY THE NUMBER OF ITERATIONS EXECUTED BY THE SOLVER (# IT.); THE LAST PHASE CORRESPONDS TO THE GLOBAL MINIMIZATION.

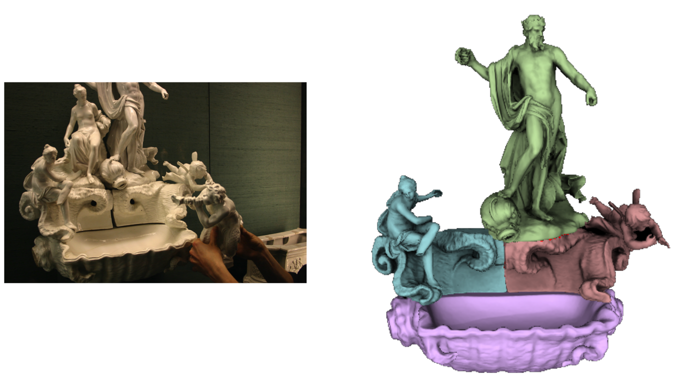

Fig. 10. The Meissen fountain.

The second case study was the Meissen Fountain, a table ornament made of hard-paste porcelain, from the Victoria and Albert (VAM) Museum (London, UK, [19]) , made of five pieces. This artwork is composed by a large set of independent pieces and represents a fountain in the grounds of Count Bruhl's Dresden palace. Some of the original pieces and the result of a partial reconstruction are presented in Figure 10. This artifact is not an example of a fractured objects; conversely, it was originally designed as a considerably large assembly of several pieces, which were supposed to be attached each others, to be used as a scenic decoration of a noble table. A few pieces are also fragmented in pieces (due to accidents incurred in their long life).
The goal of the VAM curators is to study digitally the plausible dispositions of the many pieces that form the table fountain (since these are too delicate and fragile to work with them in a physical recombination rehearsal). In this test case, the work has to be driven by a scholar/curator, since we do not have a fragmented object with matching fracture surfaces. The corresponding surfaces of adjoining portions have been produced with different molds and a manual process, therefore we cannot rely on the automatic detection of relevant and corresponding features (since there is no fractured region on the surface). This is an ongoing project, since VAM has 3D scanned only around one fourth of the pieces so far; some preliminary results of the reassembling process, obtained on the subset of pieces we have in digital format, is shown in Figure 10.

The last example we present is a laboratory experiment (see Figure 11) that we include for comparison purpose with one of the automatic recombination approaches. The fragmented object is the same used by [6] in their experiments and kindly distributed by the authors of that paper. Our user needed about 35 minutes to compete the reassembly process. This example makes it clear how much the final global minimization step allows to improve the convergence to a better, more regular, recombination.

VI. Conclusions and Future Work

We introduced a semi-assisted approach for reassembling fractured or composed objects. Our system has been designed to involve the user in the reconstruction loop, exploiting its experience and knowledge. The idea is to offer a number of constraints that the user can instantiate to define how the artifact fragments or portions should recombine. We have implemented the overall framework, building an interactive tool that incorporates only a single type of constraint, to test the feasibility, the efficiency of the constraints-based solver and to produce some preliminary results on the effectiveness and acceptance by the users of this approach. Concerning the latter objective, we successfully tested our framework with a few real application scenarios. We also demonstrated that our system is robust to very eroded or missing pieces (see Figure 8), that is a result hard to accomplish with other alignments approaches that rely only on geometrical criteria.

Our approach has been designed for being extensible and for offering different types of constraints, to go beyond the *adjacency* constraint incorporated in the current prototype.

A prototype of the proposed framework is available at http://vcg.isti.cnr.it/~pietroni/reassebly/index.html.

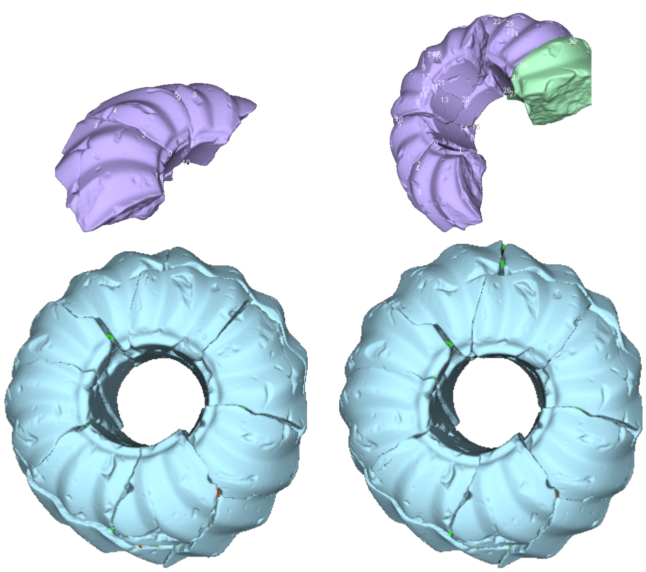

Fig. 11. A laboratory experiment based on the same model used in [6]. It is important to note how the fragments converge to a more regular shape after a global minimization step is performed.

Acknowledgments

The authors would like to thank the Pietranico Madonna restoration equipe and the colleagues at the Photographic Department at Victoria & Albert Museum (http://www.vam.ac.uk/index.html). The research leading to these results was co-funded by EU FP7 projects IST IP "3D-COFORM" (http://www.3d-coform.eu/ , G.A. no. 231809) and INFRA "ARIADNE" (http://www.ariadne-infrastructure.eu/ , G.A. no. 313193).

References

[1] L. Arbace, S. Elisabetta, M. Callieri, M. Dellepiane, M. Fabbri, I. I. Antonio, and R. Scopigno, "Innovative uses of 3d digital technologies to assist the restoration of a fragmented terracotta statue," *Journal of Cultural Heritage*, vol. 14, no. 4, pp. 332–345, July-Aug. 2013. [Online]. Available: http://vcg.isti.cnr.it/Publications/2013/AECDFAS13

[2] F. Kleber and R. Sablatnig, "A survey of techniques for document and archaeology artefact reconstruction," in *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, ser. ICDAR '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 1061–1065. [Online]. Available: http://dx.doi.org/10.1109/ICDAR.2009.154

[3] A. R. Willis and D. B. Cooper, "Computational Reconstruction of Ancient Artifacts," *IEEE Signal Processing Magazine*, vol. 25, no. 4, Jul. 2008.

[4] H. Pottmann, S. Leopoldseder, and M. Hofer, "Simultaneous registration of multiple views of a 3d object," in *Intl. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXIV, Part 3A, Commission III*, 2002, pp. 265–270.

[5] S. Krishnan, P. Y. Lee, J. B. Moore, and S. Venkatasubramanian, "Global registration of multiple 3d point sets via optimization-on-a-manifold," in *SGP '05 Proceedings of the third Eurographics symposium on Geometry processing*. Eurographics, 2005.

[6] Q.-X. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann, "Reassembling fractured objects by geometric matching," *ACM Transactions on Graphics*, vol. 25, no. 3, Jul. 2006.

[7] C. Toler-Franklin, B. Brown, T. Weyrich, T. Funkhouser, S. Rusinkiewicz, and D. Texture, "Multi-Feature Matching of Fresco Fragments," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 6, 2010.

[8] B. J. Brown, L. Laken, P. Dutré, L. V. Gool, S. Rusinkiewicz, and T. Weyrich, "Tools for virtual reassembly of fresco fragments," in *International Conference on Science and Technology in Archaeology and Conservations*, Dec. 2010.

[9] D. R. Koller, "Virtual archaeology and computer-aided reconstruction of the severan marple plan," in *Beyond Illustration: 2D and 3D Digital Technologies as Tools for Discovery in Archaeology British Archaeological Reports International Series*. Archaeopress, 2008, pp. 125–134.

[10] J. C. McBride and B. B. Kimia, "Archaeological Fragment Reconstruction Using Curve-Matching," in *2003 Conference on Computer Vision and Pattern Recognition Workshop*. Ieee, Jun. 2003.

[11] S. Winkelbach and F. M. Wahl, "Pairwise matching of 3d fragments using cluster trees," *Int. J. Comput. Vision*, vol. 78, no. 1, pp. 1–13, Jun. 2008. [Online]. Available: http://dx.doi.org/10.1007/s11263-007-0121-5

[12] C. Sanchez-Belenguer and E. Vendrell-Vidal, "Archaeological fragment characterization and 3d reconstruction based on projective gpu depth maps," in *Virtual Systems and Multimedia (VSMM), 2012 18th International Conference on*, 2012, pp. 275–282.

[13] G. Papaioannou, E.-A. Karabassi, and T. Theoharis, "Virtual archaeologist: Assembling the past," *IEEE Comput. Graph. and Appl.*, vol. 21, no. 2, Mar. 2001.

[14] D. Parikh, R. Sukthankar, T. Chen, and M. Chen, "Feature-based Part Retrieval for Interactive 3D Reassembly," in *Applications of Computer Vision, 2007. WACV'07. IEEE Workshop on*. IEEE, 2007.

[15] N. Mellado, P. Reuter, and C. Schlick, "Semi-automatic geometry-driven reassembly of fractured archeological objects," in *Proc. VAST 2010*. Eurographics, 2010, pp. 33–38.

[16] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *Proc. of Int. Conf. on 3D Digital Imaging and Modeling*. IEEE, 2001.

[17] K. Pulli, "Multiview registration for large data sets," in *Proc. of Int. Conf. on 3-D Digital Imaging and Modeling*. IEEE, 1999, pp. 160–168.

[18] Visual Computing Lab, ISTI - CNR, "Meshlab," http://meshlab.sourceforge.net/.

[19] Victoria & Albert Museum, "Meissen fountain - Victoria & Albert Museum," http://collections.vam.ac.uk/item/O10640/fountain/.