Semi-automatic geometry-driven reassembly of fractured archeological objects

Nicolas Mellado, Patrick Reuter, Christophe Schlick

The 11th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST (2010)
A. Artusi, M. Joly-Parvex, G. Lucet, A. Ribes, and D. Pitzalis (Editors)

Semi-automatic geometry-driven reassembly of fractured archeological objects

N. Mellado[1] and P. Reuter[1] and C. Schlick[1]

[1]Universités de Bordeaux, INRIA Bordeaux Sud Ouest - CNRS

Abstract

3D laser scanning of broken cultural heritage content is becoming increasingly popular, resulting in large collections of detailed fractured archeological 3D objects that have to be reassembled virtually. In this paper, we present a new semi-automatic reassembly approach for pairwise matching of the fragments, that makes it possible to take into account both the archeologist's expertise, as well as the power of automatic geometry-driven matching algorithms. Our semi-automatic reassembly approach is based on a real-time interaction loop: an expert user steadily specifies approximate initial relative positions and orientations between two fragments by means of a bimanual tangible user interface. These initial poses are continuously corrected and validated in real-time by an algorithm based on the Iterative Closest Point (ICP): the potential contact surface of the two fragments is identified by efficiently pruning insignificant areas of a pair of two bounding sphere hierarchies, that is combined with a k-d tree for closest vertex queries. The locally optimal relative pose for the best match is robustly estimated by taking into account the distance of the closest vertices as well as their normals. We provide feedback to the user by a visual representation of the locally optimal best match and its associated error. Our first results on a concrete dataset show that our system is capable of assisting an expert user in real-time during the pairwise matching of downsampled 3D fragments.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms I.3.6 [Computer Graphics]: Methodology and Techniques — Interaction techniques

1. Motivation

Archeological objects are often broken and fractured into a large amount of fragments, and the real-world reassembly of the fragments is sometimes impossible due to their size, weight, fragility, or inaccessibility. With the increasing availability of 3D laser scanners, the scanning of fragments is becoming more and more popular, resulting in large collections of detailed fractured archeological 3D objects (see for example [TFK*09, Lev00]). Fragments can then be reassembled virtually, and gained information used as a blueprint to reconstruct the real-world object.

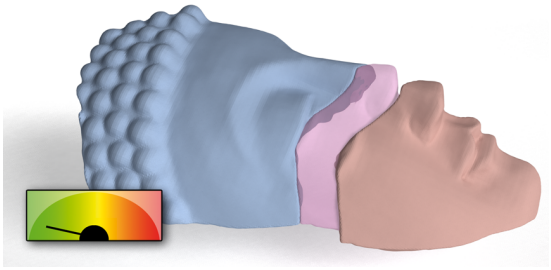

Figure 1: *Pairwise reassembly*

Let us first recall some previous work about virtual reassembly. On the one hand, there is a variety of techniques to fully *automatically* reassemble fractured objects. Most of these techniques are based on a pairwise matching of geometric features, for example based on the iterative closest point algorithm (ICP) [HFG*06, BTFN*08], or based on estimates of axis/profile curves [WOC03, KS04]. Unfor-

tunately, purely automatic reassembly methods are highly data-dependent and generally fail when fragments are missing or strongly deteriorated.

On the other hand, fragments can be reassembled *manually* on the computer by archeologists, through the help of an efficient computer-human interface, thus taking into account human expertise and knowledge. To overcome the difficult problem of positioning and orienting 3D models relative to each other, a bimanual tangible user interface for the efficient reassembly of fractured archeological objects has been presented before. It allows the user to reassemble two fragments as if they were in his hands [RRC*07]. Unfortunately, purely manual reassembly lacks precision, and does not benefit from a computer assistance, although the geometry is available.

Our contribution consists in a semi-automatic pairwise fragment matching technique that takes into account both the archeologist's expertise, as well as the power of automatic geometry-driven matching algorithms. Previous work on semi-automatic reassembly has not taken into account the user-specified relative positions and orientations *during* the interaction. For example, it either requires the user to specify *beforehand* corresponding landmarks [RJ04] or annotations [KTNL05] for guiding the matching, or *afterwards* to make the final choice between multiple assembly propositions [PKT01, TFK*09].

In contrast to these methods, our new semi-automatic reassembly approach integrates the user in a real-time interaction loop: an expert user specifies approximate initial relative positions and orientations between two fragments that are used by the system to "snap" them to the locally best match. Once aligned, the two fragments are virtually linked and considered as a single new fragment. The complete reassembly of a complex broken object can thus be done by successive pairwise matchings.

The contributions of this paper are threefold:

- A semi-automatic reassembly approach that integrates the user in the real-time interaction loop and steadily finds the best local match according to the user-specified initial pose,
- a bounding sphere hierarchy combined with a k-d tree for the identification of the potential contact surface between the two fragments to match as well as for efficient closest point queries, and
- an ICP-based matching method that takes into account both the influence of the positions and normals of the contact surface by using a bi-factorial weighting function, and that is robust to slight changes of the initial pose.

This paper is organized as follows. In section 2, we present the idea of an interaction loop that ensures the integration of expert user knowledge. In section 3, we show how we ensure the robust geometric matching to operate in real-time. In section 4, we present qualitative and quantitative results, before we conclude with directions to future work in section 5.

2. Semi-automatic interaction loop

The central idea of this paper is to increase the efficiency of the reassembly process by integrating real-time geometry-driven matching algorithms into the user interaction loop, as can be seen in figure 2.

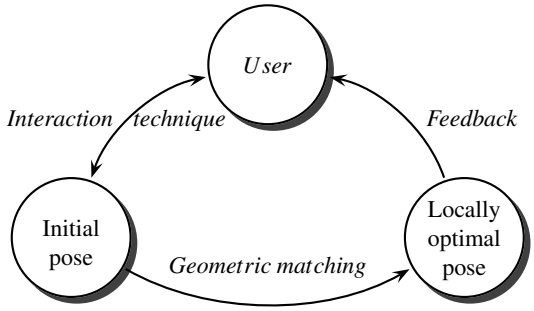

Figure 2: *Semi-automatic interaction loop.*

There are three principal prerequisites for this semi-automatic interaction loop: first, we need an efficient *interaction technique* for the specification of approximate initial relative positions and orientations between two fragments. Second, we need a real-time *geometric matching* algorithm that finds the locally optimal match with respect to this relative initial pose. Third, we have to provide a *feedback* that makes it easy for a user to validate the match, or refine the input. In the following, we show the choices that we have made for each of these prerequisites.

2.1. Interaction technique

The role of an expert in pairwise semi-automatic reassembly is to specify relative positions and orientations of the fragments that are used as input for the matching algorithm. In order to keep concentrated on the actual archeological

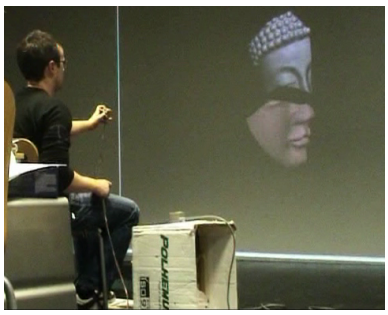

Figure 3: *The bimanual tangible user interface*

task and the involved high-level knowledge without being distracted with complex interaction techniques, we rely on

previous work dealing with bimanual tangible user interfaces [RRC*07]: as can be seen in figure 3, in each hand, the user manipulates an electromagnetically tracked prop, and the translations and rotations are directly mapped to the corresponding virtual fragments on the display. For each hand, a corresponding foot pedal is used to activate props tracking (or "clutch").

2.2. Geometric matching

We rely on an algorithm based on the iterative corresponding point (ICP) [BM92] for real-time matching, as it was already made in previous work on automatic virtual reassembly [HFG*06, BTFN*08]. Whereas the ICP is mostly used for registration to transform the surface sheets from different scans of *a single object* in a common coordinate frame, we use it for the alignment of contact surfaces for scans coming from *two different objects*. Consequently, the ICP has to be modified in order to reject all the surface parts that do not belong to the contact surface. As the name suggests, the ICP algorithm calculates the registration iteratively, and it directly works on the scanned 3D points that define the surface (in the remainder of this paper, we will refer to the scanned points as *vertices*). In each iteration, the algorithm *selects the best corresponding vertices* in order to calculate the locally optimal rotation and translation for surface alignment, by minimizing an objective function. In section 3, we explain what we understand by "best corresponding vertices" for the alignment of the contact surface, and how we select and weight them in real-time.

2.3. Feedback

Once the locally optimal match with respect to the user-specified initial pose has been determined, we have to provide feedback to the user about the locally best match. We use a visual transparent 3D representation of the matching result, as can be seen in figure 1. Based on this visual feedback, the user can evaluate if the matching result is geometrically plausible and coherent with his intent and validate the proposition or specify new initial positions and orientations. As can be seen in figure 1, we also provide a graphical indicator of the local error of the ICP algorithm, i.e. the root mean square (RMS) error of the matching of the contact surface.

Note that with this visual feedback, the reassembly "snaps" to the locally best corresponding match, like a magnet that sticks the fragments together, as long as the provided pose is within the distance threshold, and as long as it is not closer to a different local minimum. A similar "snap" metaphor has proven to be efficient in 2D vector graphics applications, where imaginary grid lines at a coarse spacing help to precisely align 2D objects despite a roughly aligned input.

3. Optimizing the geometric matching

3D models of fragments are acquired with 3D scanning devices; they are thus often noisy and consist of millions of vertices. However, the interaction loop involved in our semi-automatic matching algorithm (figure 2) requires our ICP-based matching algorithm to operate in real-time, and to robustly align the two surface sheets even in the presence of noise. According to Rusinkiewicz's fast ICP variants [RL01], an iteration is composed of six steps that are likely to be optimized: data selection, pairwise vertex matching, weighting pairs, rejecting pairs, computing an error, and minimizing the error. In the following, we show how we optimize the first four steps for efficient reassembly. More precisely, we will focus on the one hand on the vertex pair selection in order to reduce the amount of data and to fulfill the real-time constraint, and on the other hand on the weighting and rejection of the pairs for efficient and robust matching of noisy data.

3.1. Efficient vertex pair selection

For an efficient alignment of the two surface sheets, in a preprocess, we first downsample the vertices of the fragments to some tens of thousands of vertices [TL94, RL01]. For a further optimization, we only need to select the vertices that are present on the potential contact surface, while rejecting all the others. Note that in our semi-automatic reassembly method, the user specifies the relative initial positions and orientations of the two fragments so that the potential contact surface is already roughly aligned. As a consequence, we can consider that the vertices of both contact surface sheets are close to each other: their distance is less than a distance threshold d_T that can be adjusted by the user.

For the efficient detection of contact surfaces and closest vertex queries, we construct a k-d tree for each of the two fragments in a preprocess. Furthermore, inspired by classical collision detection algorithms [Lin93], we mix the k-d tree with a bounding sphere hierarchy: for every node of the k-d tree, we determine a sphere that encloses all vertices of the descendant nodes.

Consequently, in the real-time interaction loop, for every user-specified initial pose, we can efficiently determine all vertices of the contact surface by intersecting the two hierarchies recursively (see the bold spheres b_s and b'_s in figure 4). This pruning is slightly different compared to collision detection since we do not only want to obtain the sphere intersections, but also all vertices where the distance is less than the distance threshold d_T. Our solution is conservative: we increase the sphere radii by $\frac{d_T}{2}$ to capture all necessary vertices, but also vertices at a distance greater than d_T (see the dotted spheres in figure 4). To obtain the closest vertex pairs, for every vertex of the potential contact surface of the first fragment, we efficiently determine the closest vertex of the second fragment by using the k-d tree.

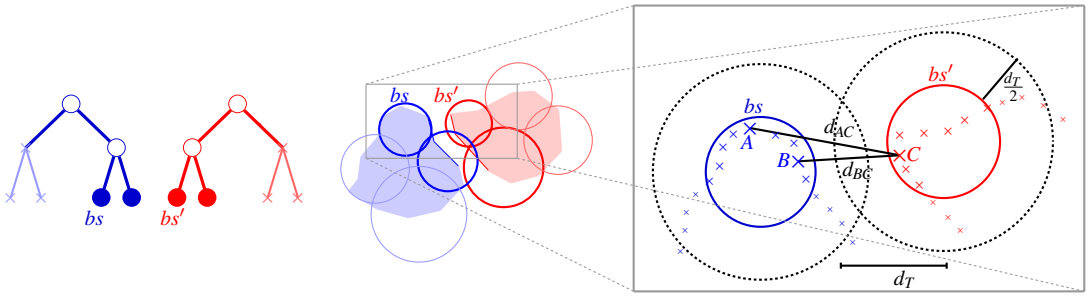

Figure 4: *(left) The two bounding sphere hierarchies. (right) Determining the contact surface by conservative pruning of the two hierarchies: even though $d_{BC} < d_T$, the sphere intersection does not detect it. By increasing the sphere radii by $\frac{d_T}{2}$, the intersection is successfully determined, but too distant vertices are detected as well: $d_{AC} > d_T$.*

3.2. Vertex pair weighting and rejection

Recall that once all the closest vertex pairs are conservatively determined, we have to reject all the pairs with a distance d greater than d_T in order to respect the contact surface. Moreover, we have to determine the influence of each pair. Following Pulli [Pul99], we take into account normal coherence in addition to vertex distance: pairs with normals of opposite direction should have a greater influence (marked in green in figure 5) compared to other neighboring pairs of the contact surface (marked in yellow in figure 5). Pairs where the dot product of the normals η is greater than a threshold η_T should not be taken into account at all. The result is a better convergence of the ICP algorithm.

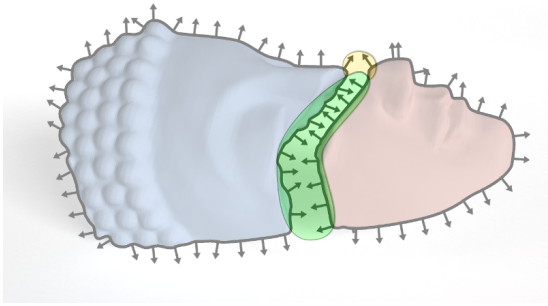

Figure 5: *The normal coherence.*

Due to the interaction loop in our semi-automatic reassembly, we must ensure that the locally optimal match evolves coherently over time. Indeed, when the user slightly changes the relative pose of the two fragments, the matching should only vary gradually in order to prevent from popping artifacts when some vertex pairs oscillate around the distance threshold d_T or the normal coherence threshold η_T. Therefore, we propose to smooth both the distance of the vertex pairs and their associated normals by the decay function $f(x) = (x^2-1)^2$. This results in the weighting functions f_d for the vertex distance d (equation 1) and f_n for the normal coherence (equation 2). A zero weight corresponds to a pair rejection.

$$\begin{cases} f_d(d) = \left(\left(\frac{d}{d_T}\right)^2 - 1\right)^2 & \text{if} \quad 0 \le d \le d_T \\ f_d(d) = 0 & \text{otherwise} \end{cases} \tag{1}$$

$$\begin{cases} f_n(\eta) = \left(\left(\frac{1+\eta}{1-\eta_T}\right)^2 - 1\right)^2 & \text{if} \quad -1 \le \eta \le -\eta_T \\ f_n(\eta) = 0 & \text{otherwise} \end{cases} \tag{2}$$

We combine both weighting criteria in the following bifactorial weighting equation, where $\alpha \in [0;1]$ is a parameter that can be tuned by the user.

$$B_f(d,\eta,\alpha) = \frac{e^{\left(-\frac{\alpha}{f_d(d)}\right)} * e^{\left(-\frac{1-\alpha}{f_n(\eta)}\right)}}{e^{-1}} \tag{3}$$

4. Results

We now show results for each of the choices we have made for the interaction technique, the matching algorithm and the feedback provided to the user. All the results are based on an informal user study, as well as on our own experiments on a variety of fractured objects that we have tried to reassemble pairwisely.

4.1. Interaction technique

Concerning the bimanual tangible user interface [RRC*07], we have observed a good acceptance and an efficient manipulation after a short learning process. However, the lack of haptic feedback and the absence of a proper collision detection is sometimes irritating during fragment manipulation. Moreover, the ICP algorithm has difficulties to converge when objects interpenetrate at the initial pose due to the high number of closest vertices that are not at the contact surface, as explained in the following section.

4.2. Geometric matching

We analyzed the results of our ICP-based geometric matching technique quantitatively and qualitatively.

Concerning the qualitative results, we first analyzed the

influence of the distance threshold d_T, the normal threshold η_T, and the parameter α that adjusts the influence of the positions and normals of the vertices of the contact surface (equation 3).

When choosing a too small distance threshold d_T or normal threshold η_T, the ICP algorithm does not have enough input points to converge to the same local optimum over time (i.e. with slight changes of the initial pose). On the contrary, when choosing too large thresholds, there are too many selected vertex pairs that do not belong to the contact surface, resulting in an inefficient and wrong matching. As a general rule, the distance threshold d_T has to be defined specifically for each fragment pair, since for the alignment of the two subparts of the fragments we cannot know in advance the size of the contact surface with respect to the size of the entire objects. Inversely, the normal threshold η_T is somehow easier to adjust: in our experiments, we have generally used $\eta_T = 0.5$ corresponding to a normal cone with a radius of 45 degrees. This threshold can further be reduced when there is less noise in the datasets.

As said above, the parameter α adjusts the influence of the positions and normals of the vertices of the contact surface. Whereas using only the distances of the vertex pairs ($\alpha = 1$) results in an inefficient matching by taking into account the interpenetrating parts of the surfaces, using only the normals ($\alpha = 0$) does not filter out surface parts that are too far away and do not spatially belong to the contact surface. As expected, taking into account both positions and normals produces better results, and we have tried several values for different datasets. In general, we experienced the best results for $\alpha \approx 0.2$ corresponding to a higher influence of the normal coherence compared to the distance.

Concerning the quantitative results, we analyzed the number of vertex pairs located in the contact zone that are feasible to process in real-time. Of course, the required time for the ICP-based matching depends on the number of iterations. Usually, the ICP converges quite rapidly (3-4 iterations) on well-defined thresholds d_T and η_T combined with a good initial pose, and we are able to deal with almost $10,000$ vertex pairs in the contact surface per second on a single core Intel Pentium 4 at 3.0Ghz running Linux. When the ICP converges too slowly or diverges, our real-time interaction loop does not wait for the result and treats the next initial pose. In order to fulfill the real-time constraint, the desired frame rate can be user-adjusted to either take into account more vertex pairs, or to ensure a faster processing and thus feedback. Recall that for scanned fractured objects with millions of vertices, we first have to downsample the objects to some tens of thousands of vertices in order to satisfy the real-time constraint. For example, both fragments of the fractured head of figure 1 were downsampled to some $15,000$ vertices.

4.3. Feedback

Concerning the *feedback* provided to the user, the informal user study shows that the visual feedback helps to analyze the position of the best match and to reason about its plausibility. Moreover, the graphical indicator allows the user to rapidly detect whether the local matching of the contact surfaces converges well. By an interpretation of both visual feedbacks, the user has a complete understanding of the global and local coherence of the matching.

5. Conclusion and future work

In this paper, we have presented a semi-automatic reassembly approach based on a real-time interaction loop. This interaction loop consists of an efficient bimanual interaction technique, a real-time matching algorithm, and a way to provide a visual feedback to the user about the best match and its associated error. As a consequence, we consider the user as the key of your approach: his knowledge and his capacity to integrate semantic knowledge in the reassembly process are used to increase the performance of the matching.

Our first results of an informal user study show that our system is capable of assisting an expert user in real-time during the pairwise matching of downsampled 3D fragments. Although our algorithm is optimized with spatial data structures, it could further be accelerated by a better exploitation of the system resources (e.g. multithreading or GPU computation). Note also that for every pair of fragments, there are three parameters that have to be adjusted beforehand which confers flexibility to the technique. However, it can sometimes be tedious to correctly choose these parameters, especially for noisy and eroded objects. This is a direct consequence of using the ICP algorithm for matching, as it only takes into account positions and normals.

In future work, we plan to integrate higher order derivatives in the semi-automatic reassembly process, as for example curvatures that we can represent by second order polynomials. We also believe that an a priori analysis of the entire fragment for salient feature detection at different scales could be used to identify potential matching candidates and their initial relative poses that can then be validated by expert users by means of the visual feedback. This first selection could reduce the number of potential matching candidates and thus makes sequentiel fragment matching more efficient.

Acknowledgments

3D models are courtesy by Vienna University of Technology. This work was supported by the ANR SeARCH project, grant ANR-09-CORD-019 of the French Agence Nationale de la Recherche.

References

[BM92] BESL P., MCKAY N.: A method for registration of 3-d shapes. IEEE Trans. on Pattern Analysis and Machine Intelligence 14, 2 (1992), 239–256. 3

[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 Theran wall paintings. ACM Trans. Graphics (Proc. SIGGRAPH) 27, 3 (Aug. 2008). 1, 3

[HFG*06] HUANG Q.-X., FLÖRY S., GELFAND N., HOFER M., POTTMANN H.: Reassembling fractured objects by geometric matching. ACM Trans. Graphics 25, 3 (2006), 569–578. 1, 3

[KS04] KAMPEL M., SABLATNIG R.: On 3d mosaicing of rotationally symmetric ceramic fragments. ICPR 2 (2004), 265–268. 1

[KTNL05] KOLLER D., TRIMBLE J., NAJBJERG T., LEVOY N. G. M.: Fragments of the city: Stanford's digital forma urbis romae project. In Proc. of the Third Williams Symposium on Classical Architecture, Journal of Roman Archaeology suppl. (2005). 2

[Lev00] LEVOY M.: Digitizing the forma urbis romae. In PreProc. of the SIGGRAPH/Eurographics Workshop on Interpreting the Past (2000), pp. 31–36. 1

[Lin93] LIN M. C.: Efficient collision detection for animation and robotics. PhD thesis, University of California, Berkeley, 1993. Chair-Canny,, John F. 3

[PKT01] PAPAIOANNOU G., KARABASSI E.-A., THEOHARIS T.: Virtual archaeologist: Assembling the past. IEEE Computer Graphics and Applications 21, 2 (2001), 53–59. 2

[Pul99] PULLI K.: Multiview registration for large data sets. In Proc. 3DIM (1999). 4

[RJ04] R.A. JOHNSON K. BARNES T. L.-S. . N. P.: Use of a hand-held laser scanner in palaeontology: A 3d model of a plesiosaur fossil. Image and Vision Computing (2004). 2

[RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the icp algorithm. In Proc. of the Third Intl. Conf. on 3D Digital Imaging and Modeling (2001), pp. 145–152. 3

[RRC*07] REUTER P., RIVIÈRE G., COUTURE N., SORRAING N., ESPINASSE L., VERGNIEUX R.: Archeotui - a tangible user interface for the virtual reassembly of fractured archeological objects. In Proc. of VAST 2007 (2007), Eurographics. 2, 3, 4

[TFK*09] THUSWALDNER B., FLÖRY S., KALASEK R., HOFER M., HUANG Q.-X., THÜR H.: Digital anastylosis of the octagon in ephesos. J. Comput. Cult. Herit. 2, 1 (2009), 1–27. 1, 2

[TL94] TURK G., LEVOY M.: Zippered polygon meshes from range images. In Proc. of SIGGRAPH 94 (1994), ACM, pp. 311–318. 3

[WOC03] WILLIS A., ORRIOLS X., COOPER D. B.: Accurately estimating sherd 3d surface geometry with application to pot reconstruction. Computer Vision and Pattern Recognition Workshop 1 (2003), 5. 1