Interactive Reconstruction of Archaeological Fragments in a Collaborative Environment

Yifan Lu*
Information Engineering, RSISE, CECS
Australian National University
Yifan.Lu@rsise.anu.edu.au

Henry Gardner†
Computer Science, FEIT, CECS
Australian National University
Henry.Gardner@anu.edu.au

Huidong Jin‡
National ICT Australia
Huidong.Jin@nicta.com.au

Nianjun Liu§
National ICT Australia
Nianjun.Liu@nicta.com.au

Rhys Hawkins¶
ANU Supercomputing Facility
Australian National University
Rhys.Hawkins@anu.edu.au

Ian Farrington‖
Archaeology and Anthropology, CASS
Australian National Univerisity
Ian.Farrington@anu.edu.au

Abstract

The automatic reassembling of archaeological artefacts from a collection of fragments is a crucial problem in archaeology. It is arduous and time-consuming because the available information, in the form of fragments, is limited and "noisy". Previous research to assist in reassembly of artefacts has largely focused on either pattern-recognition or augmented-visualisation based perspectives. This paper presents a computer-aided and collaborative system for the reconstruction of archaeological artefacts, using boundary-matching estimation by string registration. The system has three key components. It uses invariant features to represent the 3D boundary curves of fragments. It utilises robust string matching to search the globally optimal alignment so as to tolerate noise. To further handle limited and noisy information, it creates a collaborative environment to allow multiple archaeologists to remotely reassemble artefacts at the same time. A series of experiments verify the acceptable performance of the system as well as its components.

1. Introduction

A typical archaeological study involves archaeologists travelling to various archaeological sites to unearth interesting and valuable artefacts. In order to analyse these artefacts, it is essential to classify and reconstruct complete artefacts from a potentially huge number of collected fragments and most archaeological studies are still manually completed in the field or a laboratory. Archaeologists need to classify thousands of pieces of fragments into hundreds of categories, and then find the fragments that originate from a specific artefact within a category. Following classification, a subsequent major challenge is the reassembly of fragments to restore the original artefacts. Commonly, archaeological fragments have been deformed as well as broken. They are often of similar texture and appearance. Such difficulties exacerbate the classification and reassembly problems.

Previous work has considered computer assistance for the reassembly problem. For example, Willis and Cooper [12] have presented a comprehensive framework for automatically reassembling 3D pots given 3D measure-

*Department of Information Engineering, Research School of Information Science and Engineering, College of Engineering and Computer Science, Australian National University, Canberra, ACT 0200, Australia

†Department of Computer Science, Faculty of Engineering and Information Technology, College of Engineering and Computer Science, Australian National University, Canberra, ACT 0200, Australia

‡National ICT Australia (NICTA), Locked Bag 8001, Canberra ACT 2601, Australia

§National ICT Australia (NICTA), Locked Bag 8001, Canberra ACT 2601, Australia

¶ANU Supercomputing Facility, Australian National University,Canberra, ACT 0200, Australia

‖School of Archaeology and Anthropology, College of Arts and Social Sciences, Australian National University, Canberra, ACT 0200, Australia

0-7695-3067-2/07 $25.00 © 2007 IEEE
DOI 10.1109/DICTA.2007.49

ments of fragments assuming that all of the pots have a symmetric axis. Kong and Kimia [7] provided an automated method for 2D and 3D "jigsaw-puzzle" solving. Their algorithm had two stages: local shape-matching followed by global search and reconstruction. Papaioannou et al [9] have focused on the surface geometry. They used a global optimisation method to minimise an error measurement of the complementary matching between two object parts at a given relative pose, based on a point-by-point distance between the mutually-visible faces of the objects. However its performance relies on having highly-detailed or densely-sampled models and it also suffers from computational complexity. Kampel et al [6] started with the estimation of the correct orientation of the fragment, which led to the exact position of a fragment on the original vessel, and then classified the fragments based on their profile section. As the orientation of the candidate fragments was already known, the alignment of two fragments could be achieved in a two-degrees-of-freedom search space. They proposed a matching algorithm based on the point-by-point distance between facing outlines. Recently, Benko et al [1] discussed a visual interaction system for archaeology that was introduced to establish an experimental, collaborative, mixed-reality system for allowing multiple users to do off-site simulation of archaeological excavation.

Existing approaches require either the use of additional information or strong constraints on the nature of the artefact. In this paper, we propose an approach from a different perspective: Because humans have some complementary and superior capabilities to computers in making a perceptual selection based on prior experience and knowledge, it seems advantageous to combine both interactive and automatic approaches in one application.

Our approach illustrated in Figure 1 works as follows: In the first step, fragments are photographed using a high-quality digital camera from different angles. In the second step, photogrammetry software is used to produce 3D VRML models of the fragment and to extract the associated boundary curves. The 3D VRML models are then imported to our application and the boundary curves are transformed into curvature and torsion form as described in Section 2. In the next step, one boundary curve represented by curvature and torsion is matched with others using Cyclic Edit Distance algorithm as described in Section 3. A ranked list in order of descending matching likelihood is generated by comparing one target boundary curve with other boundary curves. An archaeologist can view the fragments and select one boundary curve of interest from a fragment and find out which other fragments have a high probability of originating from the same artefact based on both the automatic ranking and their own expertise. Our application, enhanced by deployment onto the Access Grid [3], allows several archaeologists to interactively reassemble artefacts together

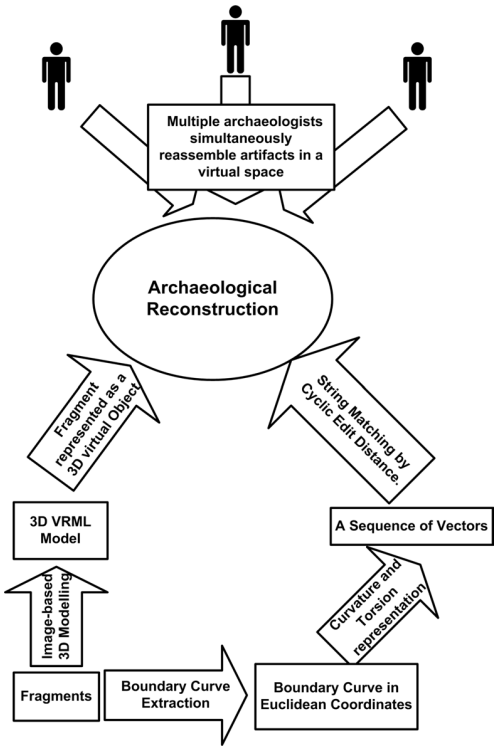

Figure 1. The Conceptual Architecture

as described in Section 4.

This paper is organised to follow the sequence of steps in the reconstruction problem described above. Experimental results, conclusion & future work are presented in Sections 5 and 6.

2. Invariant Features: Curvature and Torsion

The amount of detail included in fragment representation greatly impacts on the performance of computer-aided archaeological reconstruction. A detailed fragment representation can dramatically increase the time and space complexity of the reconstruction problem [9]. On the other hand, insufficient sample representation can cause an inaccurate or incorrect result. Our approach is to concentrate on the boundary curves of the breaking faces of fragments because they contain the most critical information for reassembly.

Data from fragments acquired from image-based modelling are represented in different local coordinate systems. The estimation and comparison of this data cannot be conducted without coordinate transformation, a non-trivial task which introduces an extra computational load. For instance, a regular ICP algorithm [2] cannot be used in this situation. One way of eliminating this problem can be found in the

local theory of curves from differential geometry. This theorem states that any two curves which have identical curvature and torsion are the same curve regardless of translation and rotation.

We employ the curvature and torsion representation to describe a boundary curve of breaking faces. As a result, a string registration technique can be applied to compare boundary curves. Curvature and torsion of a regular parameterised curve, $\overrightarrow{\mathcal{C}}$, are defined as:

$$\begin{aligned} \kappa &= \left|\overrightarrow{\mathcal{C}}''\right|, \\ \tau &= \frac{1}{\kappa^2}\left[\overrightarrow{\mathcal{C}}'\overrightarrow{\mathcal{C}}''\overrightarrow{\mathcal{C}}'''\right], \text{ where} \\ \left[\overrightarrow{a}\,\overrightarrow{b}\,\overrightarrow{c}\right] &= \begin{vmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \end{vmatrix} \end{aligned}$$

In order to calculate the curvature and torsion which involve second and third order derivatives, respectively, we apply two linear regression techniques, Least Squares Interpolation and Essential, Non-Oscillatory Interpolation, to construct piecewise polynomials describing the curves that are at least third-order differentiable.

2.1. Linear Least Squares Interpolation

The linear least squares technique is a mature mathematical approach to minimising the residuals in over-sampled data. Since the most critical features are hidden in the shape of the boundary curve, the linear least squares method is intended to capture the variation of the boundary curve as accurately as possible.

In mathematical terms, the aim is to find a solution for the linear equation: $\mathbf{Ax} \approx \mathbf{b}$. To minimise the Euclidean norm squared of the residual, we need to solve the normal equation $\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$. If the matrix $\mathbf{A}^T\mathbf{A}$ is invertible, there exists an unique solution can be found from $\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$. The pseudo inverse is then calculated by a matrix factorisation method.

2.2. Essential, Non-Oscillatory Interpolation (ENO)

When constructing interpolating polynomials to represent boundary curves, there is a tradeoff between having a better fit and having a smooth, well-behaved fitting function. This is particularly difficult near a sharp edge, or corner in a fragment. The more data points that are calculated in the interpolation, the higher the degree of the resulting polynomial, resulting in greater oscillations between data points. A high-degree interpolation may be a poor predictor of a function between points, even though the accuracy at the data points will be "perfect". Essential Non-Oscillatory (ENO) Interpolation schemes [4] have been introduced as one way to handle this problem. The principle for ENO schemes is neighbouring discontinuities; the smoothing is always from the side not containing the discontinuity. The essential idea is to select from two contiguous sets of data points for interpolation the one which gives the lower variation. For example to find the polynomial approximation between the grid points x_i and x_{i+1}, a second-order polynomial can be constructed by adding either x_{i-1} or x_{i+2}, whichever produces the smoother polynomial that has a lower coefficient for the highest order term. A third-order polynomial is interpolated by choosing an additional data point, and so on.

3. Fault-Tolerant String Matching by Cyclic Edit Distance

By regarding the sequence of curvature and torsion vectors as a string, the problem of matching the 3D boundary curves can be simplified to matching two cyclic strings. Because the contour of a fragment typically has eroded, and because a fragment is often incomplete because tiny pieces are not collectable (or destroyed in collection or simply not found) a suitable string-matching algorithm should not only tolerate mis-matching an individual character but also omitting a portion of the string. Cyclic edit distance, proposed by Marzal et al [10, 5], has the ability to measure the similarity of two arbitrary strings that satisfy this criterion. Our approach uses these advantages to estimate the similarity of two cyclic strings in a robust manner.

3.1. Branch and Bound Algorithm on Cyclic Edit Distance

Considering strings x and y, Jiménez and Marzal's Branch and Bound algorithm derived from work presented in [8, 11] introduces a range in the search space that is a pair (i, j), satisfying $0 < i < j < |x|$, where $|x|$ denotes the length of string x. Initially, the algorithm computes the edit distance $d(x, y)$, assigns the edit path P_0 and $P_{|x|}$ and initialises the set of live search ranges, S, to $(0, |x|)$. S is a priority queue sorted by the value of g, a function that computes a lower bound on the edit distance. On each iteration, comparison between a suboptimal edit distance and a minimum lower bound suggests whether there are still some ranges leading to a smaller edit distance. If so, the best range (i, j), associated with the minimum lower bound, is divided into (i, k), (k, j) and a singleton k, where $k = i + \lceil (j - i)/2 \rceil$. Then it computes the current edit distance between the P_i and P_j edit paths. The minimum of prior and current edit distances is stored as the current, suboptimal edit distance. Finally, to infer whether or not (i, k) and (k, j) lead to a smaller edit distance, they need to

be added to S, the lower bounds associated with (i,k) and (k,j) are compared with the suboptimal edit distance.

Let $\sigma_i(x)$ denote the cyclic string derived when an original cyclic string x is shifted by i places, C_i denote the cost of the insertion operation, and C_d denote the cost of the deletion operation. A lower-bound function is given as:

$$g(i,k) = \frac{d(\sigma_i(x),y) + d(\sigma_k(x),y) - (C_i + C_d)(k-i)}{2}$$

Since every time (i,j) is split into two subproblems (i,k) and (k,j), in the worst-case each iteration should be executed $\log|x|$ times. On the other hand, edit distance can be achieved with $O(|x||y|)$. So the total complexity is $O(|x||y|\log|x|)$.

3.2. Kth Shortest Paths on Cyclic Edit Distance

Peris et al [5] devised a new algorithm that finds the exact edit distance in a modified edit graph by using Kth shortest paths. The algorithm meliorates the performance from $O(|x||y|log(|x|))$ to $O(|x||y| + K(|x| + |y|))$.

The cyclic edit distance algorithm establishes an edit graph for the first string and the second string concatenated with itself, but a cyclic edit graph is formed in a slightly different manner compared with a common edit graph.

Let us take an example of matching the cyclic string a="bbccacaab" with b="aabbcc". Let $|a|$ denote the length of string a and $D(i,j)$ denote the element at the ith row and jth column in the edit graph. The cyclic edit distance works with a matrix with a dimension defined by the concatenation of second string with the first, in this case $|a|+1$ by $2|b|$. Secondly, an extra source node and sink node need to be added into the edit graph, the source node should be linked against $D(0,j)$ with a directed edge of cost 0, where $j \in \{k|0 \leqslant k \leqslant |b|\}$ and similarly the sink node should be linked against $D(|a|+1, j')$ with directed edge of cost 0, where $j' \in \{k||b| \leqslant k \leqslant 2|b|\}$. For this edit graph, the Kth shortest path algorithm initially finds the shortest path by using the common shortest path algorithm. Then it enumerates all paths, starting from the shortest path, in order of ascending cost-of-edit distance until the first one which meets all criteria is found. These criteria can be interpreted assuming that an edit path is an exact edit path if it crosses $|a|+1$ rows and $|b|$ columns.

4. Collaborative and Interactive Reconstruction

Many archaeological studies require archaeologists to travel around diverse historic sites to collect data and artefacts, and archaeologists work in a vast number of different institutions worldwide. Consequently, archaeologists may have difficulties in explaining and sharing their experiences using traditional methods of communication, as it is incapable of effectively presenting archaeological data. As a consequence, it is difficult for archaeologists to collaborate over long distances. Using the Access Grid, our approach provides a virtual "space" over a multicasting network for archaeologists to more descriptively and vividly represent archaeological data in a collaborative environment.

The Access Grid is a new generation media communication platform, that is able to organise human communication and associated information in a form-free style that features customisable implementation [3]. A shared application on the Access Grid is employed for our application, to enable communicating with the Access Grid, and eventually enable audio and video communication as well as data transmission over a multicasting network. The Application Service, created in the Venue (which is an autonomous Server in an Access Grid) when a session starts, is utilised for maintaining state data. Each client joining the session will then be able to load the data and perform necessary updates. Also, to synchronise participants during the application session, clients should send events to notify any changes they make to the state. According to Access Grid architecture, when an application session is started, an Application Factory creates an Event Channel dedicated to transmission of messages between Application Clients. It also starts a web service for the application object in the Venue. This Application Service is intended for client registration and storage of necessary state information. In addition, the Event Channel and the Application Service enable the Venue to provide a mechanism for discovery, coherence and synchronisation among application clients. The Application Client implements the ShareAppClient Interface to communicate with the Venue. In this way, the shared application enhances collaboration, allowing two or more people to view, modify, and add information simultaneously.

5. Experimental Results

In order to evaluate the proposed approach regarding precision, speed and overall performance, several experiments have been conducted concerning the curvature and torsion calculation, cyclic edit distance calculation and collaborative reconstruction. Our experiments were setup on a machine with a Intel Pentium 4 CPU 3.4GHz, 1 GB of RAM memory ,Windows XP operating system and JDK 1.5.

5.1. Performance of Curvature and Torsion Calculation

Both the ENO and Least Squares methods are intuitively approximate approaches that substitute differentials by dif-

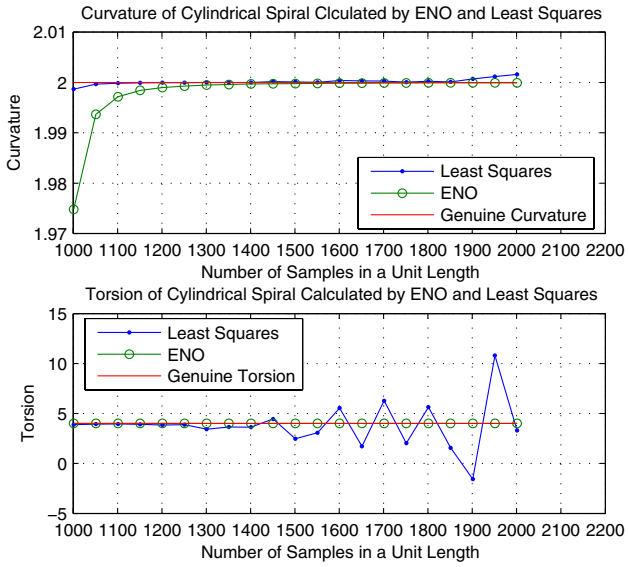

Figure 2. Accuracy of ENO and Least Squares methods

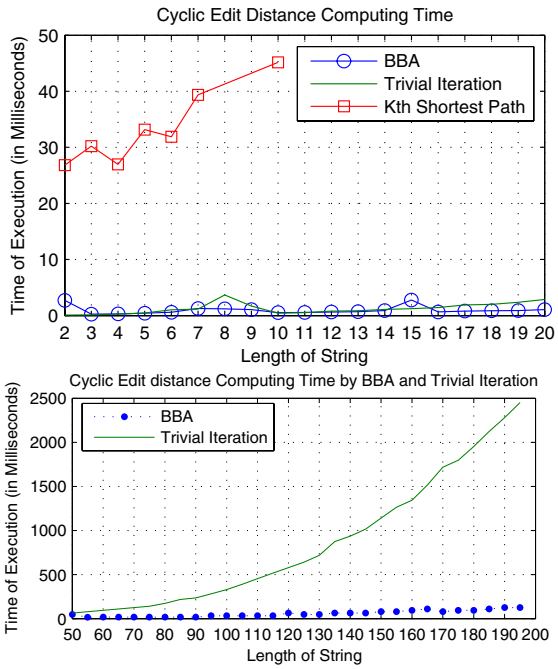

Figure 3. Speed of calculating cyclic edit distance and Speed of BBA and Trivial Iteration

ferences. Therefore, one of our experiments measured the accuracy of these approaches. In this experiment, points were sampled from a cylindrical spiral that has constant curvature and torsion:

$$\begin{aligned} x(s) &= a\cos\frac{s}{\sqrt{a^2+b^2}} \\ y(s) &= b\sin\frac{s}{\sqrt{a^2+b^2}} \\ z(s) &= \frac{bs}{\sqrt{a^2+b^2}} \end{aligned}$$

with the coefficients $a = 0.1$ and $b = 0.2$. Any curvature and torsion approximated from equal-length sample points should be close to the genuine curvature $\kappa = \frac{a}{a^2+b^2} = 2$ and torsion $\tau = \frac{a}{a^2+b^2} = 4$. A mean of curvature and a mean of torsion are compared against the genuine curvature and torsion. The result is shown as Figure 2. For curvature computations, as the number of samples per unit length increases, ENO quickly approaches the genuine curvature, while the Least Squares method displays slight fluctuation around the genuine curvature. For torsion computation, the Least Squares method appears to be a very unstable approximation to the genuine torsion: When the number of samples per unit length exceeds 1400, it suffers from large fluctuations. As expected, the ENO scheme has a far more stable estimation of torsion and it outperforms more conventional differencing methods for high order derivatives.

5.2. Performance of Cyclic Edit Distance

The computation of cyclic edit distance needs to evaluate all possible alignments between two strings. This demands a computationally competitive algorithm. Three different algorithms have been implemented, Branch and Bound Algorithm (BBA), Kth Shortest Path and the Trivial Iteration method that simply computes edit distances for all possible alignments of two cyclic strings and finds the minimum edit distance. According to a time-complexity analysis, the Kth shortest path is expected to be the fastest, while BBA should be faster than Trivial Iterative. However, our experiment indicated a different outcome: Because the Kth Shortest Path algorithm has an uncertainty in space complexity. The search space varies exponentially according to two strings' initial positions. This always causes common PC memory overflow, when the length of string (samples) more than 10. It is problematic to obtain results when the length of string (samples) is more than 10. The result is plotted as discrete symbols and shown in the first diagram of Figure 3. Generally, the Kth Shortest Path method spends much more time computing cyclic edit distance than the other algorithms. The time of execution dramatically increases for longer strings. Especially, when the length of string is equal to 8, 9 or greater 10, there are no value measured. The value of K not only affects the running time, but also determines the space complexity of $O(K|x||y|)$. Since short strings are

question	"Hard"	"Easy"	"Very Easy"
3	0	2	8
4	3	2	5
5	2	6	2
6	4	4	2
7	3	3	4
8	1	2	7

Table 1. Statistics for the first category of user questions as described in the text

question	"No"	"Maybe"	"Yes"
1	0	4	6
9	5	4	1
10	5	3	2

Table 2. Statistics for the first category of user questions as described in the text

chosen for experimental data, the BBA and Trivial Iteration algorithms cannot be clearly distinguished. In the second diagram of Figure 3 , we show the running time of BBA and Trivial Iteration for longer strings. The running time of BBA maintains relatively stable, exhibiting only a slight rise. Conversely, the running time of Trivial Iteration makes an abrupt increase as string length increases.

5.3. Experiments on Collaborative Reconstruction

Following the experiments on precision and speed, the application performance was evaluated, semi-formally, from a user perspective. Questionnaires were distributed to 10 users who were asked to complete them after they had reconstructed a model three times (a 3D model of Beethoven's bust shown in Figure 4). Questions were divided into three categories. One set of 6 questions examined the usability of specific operations on a coarse-grained Likert scale whose wording was biased in the positive direction. The questions were: "Can you load the correct model ..." (Q3), "Can you rotate the model.."(Q4), "Can you translate the model..." (Q5), "Can you observed the model from different angles..." (Q6), "Can you zoom in and out ..." (Q7), "Can you record the procedure of the reconstruction?" (Q8). The answers could be chosen from one of three categories: "hard", "easy" and "very easy". The results, shown in Table 1, were generally in the "easy" or "very easy" categories which was encouraging for the software.

A second set of 3 questions asked for subjective impressions of the software before and after use: "Do you like the interface...at first sight?" (Q1), "Do you think you ...successfully reconstructed the sculpture?"(Q9), "Do you think...is helpful software to reconstruct models?"(Q10). The answers to all these questions were taken from the categories "No", "Maybe" and "Yes" and the results are shown in Table 2. It appears from this table that first impressions were much more favourable than final, summative opinions. Comments from users which accompanied this survey indicated that some had a hard time moving models to desired positions and that some felt frustrated when they accidentally moved a correctly-placed model. These two aspects of the interface, in particular, need more work if it is to be widely used.

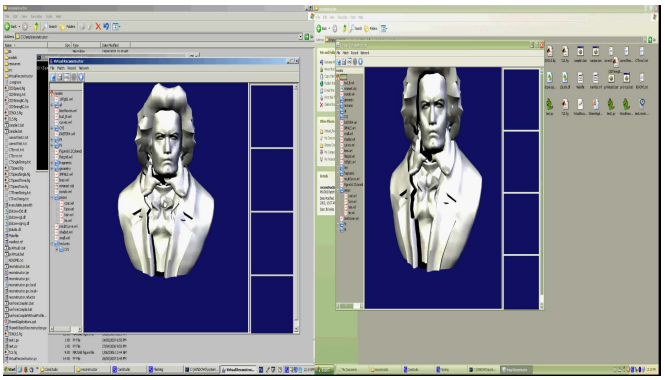

Figure 4. Two users simultaneously reassemble a 3D model of Beethoven's bust

The remaining question asked whether sufficient help could be found to explain how to use the application. All respondents responded with either "Easy" (3) or "Very Easy" (7).

6. Conclusion and Future work

With respect to a coordinate invariant representation, curve-matching by cyclic edit distance, collaborative work and interactive manipulation in three dimensions, we have constructed a framework to achieve the reconstruction of archaeological artefacts in a collaborative environment. We also have presented a comparison on the accuracy of curvature and torsion between Least Squares and ENO scheme and the performance evaluation on BBA, Kth shortest path and Trivial Iteration method. At the end, we have brought all together and verified our system in terms of user-centred evaluation. However, some aspects still have potential to be explored and are worth further investigation: The boundary curve needs to be manually marked in image-based modelling, and so an automated 3D boundary detection approach, which can automatically extract the boundary

curves from 3D model data, would be useful. Moreover, given the best match between two fragments, automatically registering two fragments in three dimensions would greatly aid archaeological reconstruction. Finally, usability aspects of the software, particularly with accurate placing of a 3D model, need to be improved.

7. Acknowledgement

Several of us would like to acknowledge useful and inspired discussions with Hongdong Li.

References

[1] H. Benko, E. W. Ishak, and S. Feiner. Collaborative mixed reality visualization of an archaeological excavation. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 132–140, 2004.

[2] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[3] L. Childers, T. Disz, M. Hereld, R. Hudson, I. Judson, R. Olson, M. E. Papka, J. Paris, and R. Stevens. ActiveSpaces on the Grid: The construction of advanced visualization and interaction environments. In *Parallelldatorcentrum Kungl Tekniska Hgskolan Seventh Annual Conference*, 1999.

[4] A. Harten. ENO schemes with subcell resolution. *Journal of Computational Physics*, 83(1):148–184, 1989.

[5] V. M. Jiménez, A. Marzal, V. Palazón, and G. Peris. Computing the cyclic edit distance for pattern classification by ranking edit paths. In *International workshop on Syntactical and Structural Pattern Recognition and Statistical Pattern Recognition*, pages 125–133, 2004.

[6] M. Kampel and R. Sablatnig. Automated segmentation of archaeological profiles for classification. In R. Kasturi, D. Laurendeau, and C. Suen, editors, *International Conference on Pattern Recognition*, pages 57–60, 2002.

[7] W. Kong and B. B. Kimia. On solving 2d and 3d puzzles using curve matching. *IEEE Conference on Computer Vision and Pattern Recognition*, 02:583, 2001.

[8] M. Maes. On a cyclic string-to-string correction problem. *Information Processing Letters*, 35(2):73–78, 1990.

[9] G. Papaioannou, E.-A. Karabassi, and T. Theoharis. Reconstruction of three-dimensional objects through matching of their parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):114–124, January 2002.

[10] G. Peris and A. Marzal. Fast cyclic edit distance computation with weighted edit costs in classification. In *International Conference on Pattern Recognition*, pages 184–187, 2002.

[11] R. A. Wagner and M. J. Fischer. The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173, 1974.

[12] A. R. Willis and D. B. Cooper. Bayesian assembly of 3d axially symmetric shapes from fragments. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 82–89, 2004.