A Multi-Scale Method for the Re-Assembly of Fragmented Objects

Helena Cristina da Gama Leitão
Inst. of Computing - Fluminense Federal Univ.
R. Passo da Pátria, 156 - 24210-240 Niterói, RJ, Brazil
hcgl@pgcc.uff.br

Jorge Stolfi
Inst. of Computing - Univ. of Campinas
Caixa Postal 6176 - 13083-970 Campinas, SP, Brazil
stolfi@dcc.unicamp.br

Abstract

We describe here an efficient algorithm for re-assembling one or more unknown objects that have been broken or torn into a large number N of irregular fragments—a problem that often arises in achaeology, art restoration, forensics, and other disciplines. The algorithm compares the curvature-encoded fragment outlines, using a modified dynamic programming sequence-matching algorithm, at progressively increasing scales of resolution. The total cost gets reduced from $\Theta(N^2L^2)$ (where L is the mean number of samples per fragment) to about $O(N^2L)$; which, in principle, allows the method to be used for problems of practical size ($N = 10^3$ to 10^5 fragments, $L = 10^3$ to 10^4 samples). The performance of the algorithm is illustrated with an artificial but realistic example.

1 Introduction

Re-assembling broken objects from a collection of thousands randomly mixed fragments is a problem that arises in several applied disciplines, such as archaeology, failure analysis, paleontology, art conservation, and so on. Solving such puzzles by hand may require years of tedious and delicate work, so the need for computer help is quite obvious.

Indeed, computers are already being used in such applications for the purpose of sorting fragments according to gross properties—color, texture, material, profile, etc.. The classification helps by reducing the number of fragment pairs that need to be compared by hand. For very large collection of undecorated fragments, however, rough classification is not enough: one needs a tool that can automatically match fragments based on their shapes — which is the problem that we address in this paper.

The difficulty of this problem lies in the large number of fragments present in typical instances. If the average fragment outline has L sample points, then finding the best fit between two fragments requires $\Theta(L^2)$ operations by standard algorithms, and $\Theta(N^2L^2)$

for a colection with N fragments. Since computer aid would be most necessary for collections with 10^3 to 10^5 pieces, and accurate matching requires 10^2–10^3 samples per fragment, exhaustive matching is not a viable option.

Here we describe an algorithm that performs this task at much lower cost, through the use of *multi-scale* techniques. As we will see, these techniques allow us to reduce the total cost from $O(N^2L^2)$ to about $O(N^2L \log L)$ [5, 2].

In a companion paper [6] we address the question of whether the problem is solvable at all. We show that, for well-preserved ceramic fragments, there is enough information in a couple of centimeters' worth of fracture line to identify the matching fragment, with acceptable accuracy, even among millions of other similar fragments.

Related work. At present, the reconstruction of archaeological fragments is done largely by hand. Computers are used, if at all, only in the enhancement, classification, and presentation of scanned images of the fragments [4], which are indexed and retrieved based solely on textual descriptions provided by the user.

Computer vision and pattern matching techniques have occasionally been used to automatically extract indexing and matching information from digital images of archaeological artifacts [13, 3, 8]. The specific problem of identifying adjacent ceramic fragments by matching the shapes of their outlines was recently considered by Üçoluk and Toroslu [13]. Their algorithm considers only a fixed scale of resolution, and therefore has large expected asymptotic cost; no real-world tests are reported in the article. Mark Levoy is investigating the use of 2D shape matching techniques for the same purpose [7].

The fragment re-assembly problem is similar to that of *automatic assembly of jigsaw puzzles*, which has been addressed as a cute exercise in robotics and machine vision. In particular, H. Wolfson and G. C. Burdea developed a program that finds matching pieces in a standard puzzle game [1], and even controls a robot arm to assemble the puzzle. However, these techniques rely on special characteristics of puzzle pieces, such as smooth borders and sharp corners, which aren't found in archaeological materials.

More generally, the problem can be viewed as a special case of *object recognition by approximate outline matching* [11]. However, in this field one typically assumes that the given outlines are to be matched against a small set of fixed templates. In our application, however, the templates are the fragments outlines themselves, which may number in the thousands. Therefore, we must discard many standard object recognition methods because they rely on extensive preprocessing of the templates.

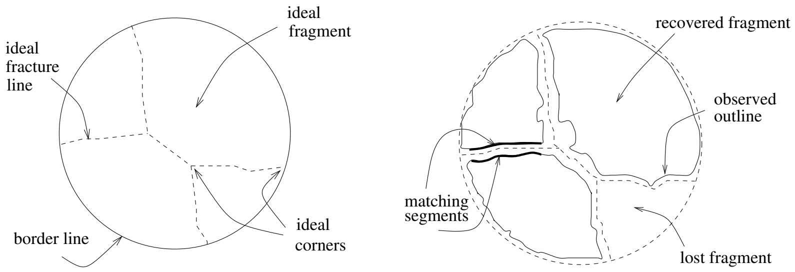
Multi-scale techniques have often been used for image-based and outline-based shape matching [9, 10, 14]. However, most prior work in this area is based on the identification of certain *critical points* of the outline, such as corners or curvature extrema. As we noted above, this approach is neither feasible nor useful in the case of ceramic fragment outlines. Therefore, our algorithm applies the multi-scale approach directly to the comparison of fragment outlines, without prior identification of critical points.

2 Statement of the problem

Fracture model. We assume that the original objects had a well defined smooth surface which was divided into two ou more parts, the *ideal fragments*, separated by *ideal fracture lines* — irregular curves of zero width. Two fragments are said to be *adjacent* if they share a fracture line. An *ideal corner* is a point where three or more fragment boundaries meet.

The boundary of an ideal fragment is an *ideal outline*; it is the concatenation of one or more fracture lines and pieces of the object's original border. See figure 1(a).

Observed outlines. Ideal fragment outlines are abstractions which cannot be determined, or even defined, with absolute accuracy. The *observed outlines* which we can extract from images of real fragments differ from the ideal outlines due to a number of reasons, either physical (such as loss of small fragments, wear, surface irregularities) or instrumental (such as parallax, shadowing, and image quantization). Thus, the ideal fracture line that separates two adjacent ideal fragments becomes two slightly different *matching segments* on the two observed fragment outlines. See figure 1(b).



(a) (b)

Figure 1: Ideal fracture network (a) and observed outlines (b).

The problem. We can now state the *fragment matching* problem as follows: given a set $\mathcal{C}$ of observed fragment outlines, identify those pairs of fragments which were probably adjacent in the original object.

3 Shape matching

Outline representation. Our algorithm assumes that each observed fragment outline is given as a circular sequence of uniformly spaced *samples* $c_0, .. c_{n-1}$, with the same sampling step δ for all outlines. Since each fragment is independently digitized in arbitrary orientation, we define the sample values as the local curvature of the outline at the sample points — a well-known shape representation which is invariant under rotations and translations of the fragments [15]. Although we have used only flat (2-dimensional) fragments in our tests, the algorithm can be used also for non-flat objects, such as ceramic and glass vessels—provided their surface is smooth enough to posess a well-defined tangent plane at each point along the fracture line.

The curvature values can be combined with any other local property of the fragment, such as color or thickness, that is invariant under rotations and translations and can be used to identify matching fragments. Since our algorithm makes no assumption about the nature of the samples, it can use such additional information whenever it is available. In the case of non-flat objects, for instance, we could (and probably should) extend each sample value with the local surface curvatures, or with the local torsion of the outline curve, as discussed by Üçoluk and Toroslu [13].

Segments and candidates. A *segment* is any sequence of one or more consecutive samples from some fragment outline. A *candidate* is any pair of segments belonging to different outlines. We say that a candidate is *true* if its segments correspond to the same ideal fracture line (or part thereof); otherwise the candidate is *false*. We denote by $\mathcal{T}$ the set of all true candidates among the given fragments. Note that on a set of N outlines with an average of L samples each we can define about NL^2 segments and N^2L^4 candidates.

Discrete pairing. The loss of material and other errors in the observed outlines may change the length of the matching segments by different amounts, so that we cannot expect a perfect one-to-one pairing of the samples. To address this problem, we must allow a flexible pairing of the samples of the two segments. We therefore define a *pairing* between the two segments $a = (a_0, .. a_m)$ and $b = (b_0, .. b_n)$—enumerated in opposite directions—as a pair (r, s) of index sequences $r_k \in \{0, .. m\}$, $s_k \in \{0, .. n\}$, $k \in \{0, .. p\}$, beginning with $(0, 0)$ and ending with (m, n), such that

$$(r_{k+1} - r_k, s_{k+1} - s_k) \ \in \ \{(0, 1), (1, 1), (1, 0)\} \tag{1}$$

for all k. We say that a pairing (r, s) establishes a correspondence between samples a_{r_k} and b_{s_k}, for all $k \in \{0, .. p\}$.

Candidate mismatch. Let (r_k, s_k), $k \in \{0, .. p\}$ be a pairing between the two segments $a = (a_0, .. a_m)$ and $b = (b_0, .. b_n)$. We define the *quadratic mismatch* of the two segments as $S^2(a, b; r, s) = D^2(a, b; r, s) + Z^2(r, s)$, where

$$D^2(a, b; r, s) \ = \ \frac{1}{4}\sum_{k=0}^{p-1}(\epsilon_k + \epsilon_{k+1})(\tau_{k+1} - \tau_k) \tag{2}$$

$$Z^2(r, s) \ = \ \frac{\zeta^2}{2}\sum_{k=0}^{p-1}|(r_{k+1} - r_k) - (s_{k+1} - s_k)| \tag{3}$$

where ζ^2 is a constant, $\epsilon_k = \|a_{r_k}, b_{s_k}\|^2$, $\tau_k = r_k + s_k$, and $\|a_i, b_i\|$ is any distance metric between the two samples values.

The term D^2 measures the total difference between the sample values of the two segments. The factor $(\epsilon_k + \epsilon_{k+1})/2$ is the mean value of $\|a_i, b_j\|^2$ along the step $(r_k, s_k) \to (r_{k+1}, s_{k+1})$ of the pairing; while $(\tau_{k+1} - \tau_k)/2$ is the mean number of sampling steps (either 1 or 1/2) spanned by that pairing step. The term Z^2 is meant to penalize pairings that are too irregular; note that Z^2 is zero if the pairing is one-to-one. This term is necessary when each sample is a single real number—since, in that case, we may sometimes obtain a very low D^2 by use of a sufficiently irregular pairing, even for segments of very different shapes. The parameter ζ^2 is the penalty for each asymmetric step (a step where only one index increases).

Critical mismatch. Intuitively, for a fixed n, the smaller the value of $S^2(a, b; r, s)$, the more likely it is that a and b are adjacent segments, and (r, s) is the true correspondence between their samples. This intuition is supported by analysis of the distribution of $S^2(a, b; r, s)$, under a simplified fracture model [2]. Assuming a fixed segment size $n = m$ with one-to-one pairing of the samples, and independent, Gauss-distributed sample values, we can derive through Bayesian analysis a critical value Ξ^2 of S^2, such that the candidate is more likely to be false if $S^2(a, b; r, s) > \Xi^2$, and true if $S^2(a, b; r, s) < \Xi^2$.

Moreover, the critical value of S^2 turns out have the form $\Xi^2 = (n - n_{\min})\xi^2$, where ξ^2 is a constant that depends only on the nature of the fragments, and $n_{\min}$ is proportional to $\log N$. The parameter ξ^2 is the *critical sample mismatch*, the mean value of $\|a_i, b_i\|^2$ that separates true candidates from false ones, provided they are sufficiently long ($n \gg n_{\min}$). The parameter $n_{\min}$ is the *minimum candidate length* required for reliable partner identification: a candidate with less than $n_{\min}$ steps is more likely to be false than true, no matter how similar its segments are.

Optimum pairing. In practice, the correct pairing (r, s) between the two segments is not one-to-one, and is not known. So we use instead the pairing (r_*, s_*) that minimizes formula (2–3), among all discrete pairings of the two segments. Moreover, the sample values are not independent. We are unable to derive the distribution of the mismatch $S^2_*(a, b) = S^2(a, b; r_*, s_*)$ analytically; however, our experiments indicate that the simplified analysis remains valid in qualitative terms. Namely, the critical value Ξ^2 of $S^2_*(a, b)$, at which the candidate is equally likely to be true or false, still varies according to $\Xi^2 = ((m+n)/2 - n_{\min})\xi^2$, for suitable costants ξ^2 and $n_{\min}$. Based on this analysis, we define the *discriminant* of a candidate as

$$\Delta(a, b) = S^2_*(a, b) - \xi^2\big((m + n)/2 - n_{\min}\big) \tag{4}$$

We then consider a candidate true if $\Delta < 0$, and false if $\Delta > 0$.

Obtaining the parameters. The parameters ξ, ζ, and $n_{\min}$ depend on sample size and on the nature of the fragments, and therefore must be empirically determined for each instance of the problem. Our method is to choose a value for ζ; compute the mismatch S^2_* for known samples of true and false candidates, of various lengths $l = (m + n)/2$; plot the values of S^2_* against l; and look for values of ξ and $n_{\min}$ such that the line $y = \xi^2(x - n_{\min})$ best separates the two samples. We then repeat this analysis for several values of ζ. (In our experiments with ceramic fragments, the best discrimination was obtained with $\zeta \approx 2.5\xi$, at all scales.)

4 The multi-scale algorithm

The comparison of two segments is expensive because the most efficient algorithm we know to compute the optimum pairing (r, s)—a variant of *dynamic programming* [12]—requires $\Theta(n^2)$ operations for segments with n samples.

To reduce this cost, we use *multi-scale* approach [9, 10, 14]. Let $L_{\min}$ be the minimum length of outline that we need to compare in order ensure a reliable decision [6]. We begin by solving the problem for coarse versions of the contours, sampled with the largest possible step δ, namely $\delta^{(K)} \approx L_{\min}$. At this scale we have only a few tens of samples per outline, so we can afford to enumerate and check all possible pairs of segments from all outlines. Of course, at this scale we cannot distinguish true candidates from mere chance resemblances, so we are left with a large set of possibly true candidates.

That done, we solve again the problem at a finer scale, with contours sampled with a smaller step $\delta^{(K-1)} = \delta^{(K)}/2$; but now we *compare only the most promising candidates found at the previous scale*. The additional details available in the higher-resolution outlines usually allow us to eliminate a large fraction of the false candidates. We repeat this process at finer and finer scales, with sampling steps $\delta^{(k)}$ decreasing in geometric progression. If the thresholds $\xi^{(k)}$ and $\zeta^{(k)}$ are properly chosen, the candidates that survive to the end of the last stage are likely to contain a large fraction of the true candidates present in the given outlines, and relatively few false ones.

Algorithm 1 *Multi-scale matching of irregular fragments.*

Inputs:

- *the raw outlines* $\mathcal{C} = \{\mathcal{C}_0, ..\mathcal{C}_{N-1}\}$*;*
- *the minimum sampling step* $\delta^{(1)}$*;*
- *the minimum length* $L_{\min}$ *for reliable matching;*
- *the parameters* $\xi^{(k)}$*,* $\zeta^{(k)}$*, and* $n^{(k)}_{\min}$*, for each scale* k*;*
- *the corner blurring factor* α*.*

Output:

- *a set* $\mathcal{S}$ *of probably true candidates.*

1. $k \leftarrow 0$; $\mathcal{C}^{(0)} \leftarrow \mathcal{C}$.
2. While $\delta^{(k+1)} \leq L_{\min} - 8\alpha\delta^{(k+1)}$, do:
   - 2.1. $k \leftarrow k + 1$.
   - 2.2. Filter and resample the outlines $\mathcal{C}^{(k-1)}$ with step $\delta^{(k)}$, obtaining outlines $\mathcal{C}^{(k)}$.
   - 2.3. $\delta^{(k+1)} \leftarrow 2\delta^{(k)}$.
3. Set $K \leftarrow k$; determine the initial candidate set $\mathcal{S}^{(K)}$ among the outlines $\mathcal{C}^{(K)}$.
4. For $k = K, K-1, \ldots, 1$, do:
   - 4.1. Refine the raw candidates $\mathcal{R}^{(k)}$, obtaining the refined ones $\mathcal{S}^{(k)}$.
   - 4.2. Remove from the set $\mathcal{S}^{(k)}$ all candidates with positive discriminant Δ.
   - 4.3. For each pair of outlines, collect all their candidates in $\mathcal{S}^{(k)}$, and eliminate all but 2^k candidates with smallest (most negative) Δ's.
   - 4.4. Merge any overlapping candidates among the remaining elements of $\mathcal{S}^{(k)}$.
   - 4.5. Map the candidates $\mathcal{S}^{(k)}$ from the outlines $\mathcal{C}^{(k)}$ to the outlines $\mathcal{C}^{(k-1)}$, obtaining the raw candidates $\mathcal{R}^{(k-1)}$.
5. Return $\mathcal{S} \leftarrow \mathcal{S}^{(0)}$.

Some points of the algorithm deserve further explanation:

Filtering and resampling. In steps 2.1–2.3 we build, for each input fragment outline $\mathcal{C}^{(0)}_i$, a set of coarsened outlines $\mathcal{C}^{(k)}_i$, $k = 1, 2, \ldots, K$, whose sampling steps $\delta^{(k)}$ increase in geometric progression. Before resampling each outline, we smooth it out by convolution with a Gaussian filter of width $\lambda^{(k)} = 4\delta^{(k)}$, ensuring that the curve can be reconstructed from the samples with negligible aliasing artifacts.

Corner blurring. A side effect of outline smoothing is that the fragment corners (points where three or more fracture lines meet) may become *blurred*, i.e. rounded off. This effect causes the ends of matching segments to pull away from each other, so that only the middle part of a true candidate can be recognized as such at the coarser scales. The distance $h^{(k)}$ where this blurring is significant depends on the comparison threshold ξ and the angle between the two fracture lines incident at the corner. In practice, we can assume that $h^{(k)} < \alpha\lambda^{(k)} = 4\alpha\delta^{(k)}$, where α is a small constant.

Coarsest and finest scales. Outline comparison starts with the curves $\mathcal{R}^{(K)}$, the coarsest version of the outlines where any true candidates with length $\geq L_{\min}$ haven't yet been obliterated by corner blurring. The stopping criterion of the filtering loop (step 2) expresses this condition, and implies that $\delta^{(K)} \leq L_{\min}/(1+8\alpha)$, i.e. $K = \log_2(L_{\min}/\delta^{(1)}) + \log_2(2/(1+8\alpha))$. The smallest useful filtering scale $\lambda^{(1)}$ is determined by the magnitude of the perturbations inherent in the observed outlines.

Initial candidates. In step 3 we solve the problem for the coarsest versions of the outlines, sampled with step $\delta^{(K)} \approx L_{\min}$. At this scale we can afford an exhaustive comparison of all pairs of segments from all outlines, and assume one-to-one pairing of samples in the two segments. So, for each pair of fragment outlines a, b, we list all pairs of maximal segments $(a_i, a_{i+1}, \ldots, a_{i+n})$ and $(b_j, b_{j+1}, \ldots, b_{j+n})$ which have $\Delta < 0$ and $n \geq L_{\min}/\delta^{(K)} - 8\alpha$; and, of each list, we keep only the 2^K candidates with smallest (most negative) Δ. The initial candidate set $\mathcal{S}^{(K)}$ is the union of these lists.

Candidate refinement. In step 4.1, we *refine* each candidate: meaning that we adjust the endpoints of its segments, and its pairing (r, s), so as to minimize its discriminant Δ, recomputed for the higher-resolution outlines $\mathcal{C}^{(k)}$. The refinement algorithm, a variant of *dynamic programming* (DP) [12], looks for a minimum cost path in a directed graph G, whose vertices are all pairs of indices (i, j) of samples a_i and b_j, respectively. The edges of G are such that any directed path corresponds to a discrete pairing (r, s) between a and b. The edge costs are assigned in such a way that the total cost of the path is the discriminant $\Delta(a, b; r, s)$, as defined by formula (4), minus the constant term $\xi^2 n_{\min}$.

Incremental matching. The DP algorithm requires $\Theta(n^2)$ operations for segments with n samples. To reduce this cost, we modified the algorithm so as to search for the minimum-cost path in G only within a band of specified width $2q$ surrounding the pairing (r', s') found at the previous scale. This change reduces the cost from $\Theta(n^2)$ to $\Theta(qn)$ [2].

Selection, pruning, and merging. In steps 4.3 and 4.4, we again discard those refined candidates with $\Delta > 0$, and keep only the 2^k best candidates for each pair of fragments; then we look for any overlapping candidates, and merge them. (Note that a true candidate C may begin as two or more candidates in the initial set $\mathcal{S}^{(K)}$, covering different parts of C with slightly different alignments. These partial candidates are likely to overlap once they get refined and mapped to finer scales.)

Candidate mapping. Step 4.5 maps each candidate of $\mathcal{S}^{(k+1)}$ — a pair of segments on the outlines $\mathcal{C}^{(k+1)}$ — to the corresponding candidate on the outlines $\mathcal{C}^{(k)}$. To account for corner blurring, we also extend the new candidate by $\alpha(\lambda^{(k+1)} - \lambda^{(k)})$ at each end.

Running time. Since we use incremental matching to process each candidate, the total cost of iteration k of the main loop (steps 4.5–4.4) is proportional to the total length of all candidates in $\mathcal{R}^{(k)}$. The length of each candidate is bounded by $c' 2^{K-k}$, for some conctant c'; and the number of candidates in $\mathcal{R}^{(k)}$ is bounded in step 4.3 by $2^k N(N-1)/2$. Therefore, the cost of each iteration is $O(2^K N^2)$. Since the number of iterations K is $\log_2 L_{\min} + O(1)$, the algorithm runs in $O(N^2 L_{\min} \log L_{\min})$ time — i.e. faster than the single-scale algorithm by a factor of $\Theta(L^2/(L_{\min} \log L_{\min}))$. (Note that $L \approx 10^3$ and $L_{\min} \approx L/10$ implies $L^2/(L_{\min} \log L_{\min}) \approx 10^3$).

Actually, the bound of $2^k N(N-1)/2$ for $\left|\mathcal{R}^{(k)}\right|$ is rather pessimistic. By Euler's theorem, the number of maximal true candidades is only $O(N)$. Therefore, if $\xi^{(k)}$, $\zeta^{(k)}$, and

$n_{\min}^{(k)}$ are appropriately chosen, the number of candidates $\left|\mathcal{R}^{(k)}\right|$ should decrease much faster than $2^k N(N-1)/2$. In that case, the total cost will be dominated by that of the first few iterations, namely $O(N^2 2^K) = O(N^2 L_{\min})$.

5 Experiments

We have coded this algorithm in Modula-3, and tested it with an artificial but realistic sample of ceramic fragments. The original objects were five rectangular unglazed ceramic tiles, about 25.0 cm by 6.0 cm, which were shattered into 112 major pieces. See figure 2.

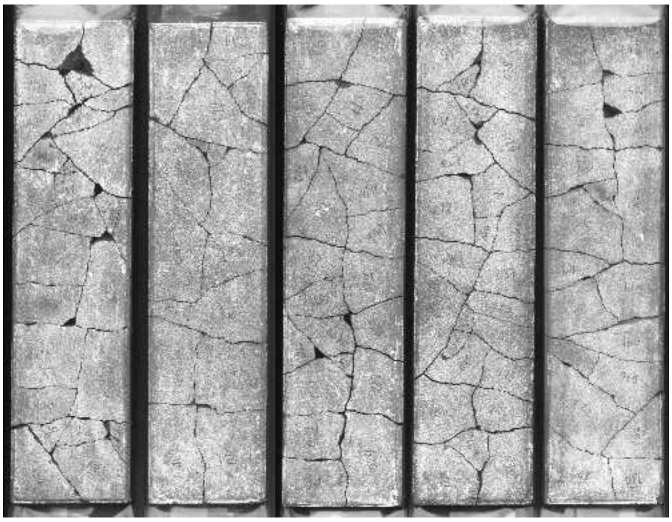


Figure 2: Test fragments, manually re-assembled.

The fragments were directly digitized with a standard flatbed scanner (UMAX model UC630 Maxcolor) at 300 pixels per inch. The flat side of each fragment was lightly rubbed with chalk to enhance contrast. The resulting images are shown in figure 3. The outlines extracted from figure 3 (whose average length L was about 1500 pixels, or 127 mm) were filtered and converted to curvature values.

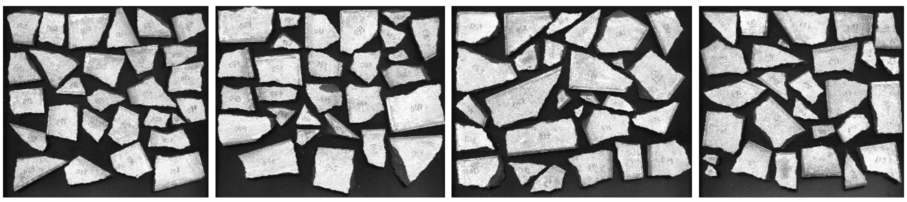


Figure 3: Input images for the test.

The multi-scale matching algorithm was then applied to these curvature-encoded outlines, beginning with scale $\lambda^{(5)} = 32$ pixels (2.7 mm) and ending with scale $\lambda^{(1)} = 2$ pixels (0.17 mm). The corner blurring factor was set to $\alpha = 6$. For each filtering scale $\lambda^{(k)}$, we show below the sample mismatch threshold $\xi^{(k)}$ and the half-step penalty $\zeta^{(k)}$ used at that scale; the number $m^{(k)}$ of candidates retained per fragment pair; the total cost of the refinement step $t_{ref}^{(k)}$ (in seconds); the number of candidates $\left|\mathcal{S}^{(k)}\right|$ after pruning and merging; and the number $\left|\mathcal{S}^{(k)} \cap \mathcal{T}\right|$ of true candidates in that set.

Table 1 below shows the parameters and results of the test. The minimum length $L_{\min}$ of candidates to look for was set at 210 pixels (17.8 mm). Note that, at the coarsest scale ($\lambda^{(5)} = 32$ pixels, $\delta^{(5)} = 8$ pixels), these candidates were reduced to $210 - 6 * 32 = 18$ pixels (≈ 3 samples). There were 74 true candidates in the input data longer than $L_{\min}$; the algorithm started with 166626 initial pairs, and returned 277 pairs, of which 46 were true. Figure 4 shows the first 40 candidates returned, in order of increasing total mismatch.

k	$\delta^{(k)}$	$\lambda^{(k)}$	$\xi^{(k)}$	$\zeta^{(k)}$	$m^{(k)}$	$\left\|\mathcal{S}^{(k)}\right\|$	$t_{ref}^{(k)}$	$\left\|\mathcal{S}^{(k)} \cap \mathcal{T}\right\|$
4	8	32	0.00080	0.00210	16	64743	31091	57
3	4	16	0.00260	0.00440	8	7797	52551	70
2	2	8	0.00900	0.01400	4	1814	10095	64
1	1	4	0.02600	0.04000	2	442	4058	50
0	0.5	2	0.07050	0.10000	1	277	2542	46

Table 1: Parameters and results of the multi-scale algorithm.

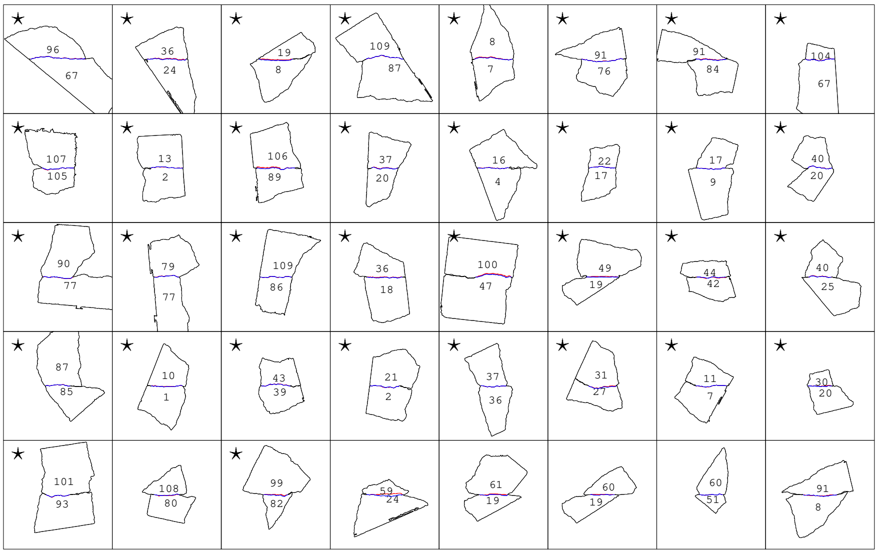


Figure 4: The first 40 of 277 candidates returned by the multi-scale matching algorithm. The stars ($\star$) identify true candidates.

5.1 Conclusions

Our experimental results, although modest in scope, demonstrate the possibility of automatically identifying adjacent fragments by matching the shapes of their outlines. They also validate the basic premise of the multi-scale matching method, namely that the false candidates are quickly eliminated as they are re-tested with increasing resolution.

References

[1] Grigore C. Burdea and Haim J. Wolfson. Solving jigsaw puzzles by a robot. *IEEE Transactions on Robotics and Automation*, 5(6):752–764, 1989.

[2] Helena Cristina da Gama Leitão. *Reconstrução Automática de Objetos Fragmentados*. PhD thesis, Inst. of Computing, Univ. of Campinas, November 1999. (In Portuguese).

[3] R. Halíř and J. Flusser. Estimation of profiles of sherds of archaeological pottery. In *Proceedings of the Czech Pattern Recognition Workshop (CPRW'97)*, pages 126–130, February 1997.

[4] Alan D. Kalvin et al. Using visualization in the archaeological excavations of a pre-Inca temple in Peru. Report RC 20518, IBM T. J. Watson Res. Center, 1996.

[5] Helena C. G. Leitão and Jorge Stolfi. Automatic reassembly of irregular fragments. Technical Report IC-98-06, Institute of Computing, Univ. of Campinas, April 1998.

[6] Helena C. G. Leitão and Jorge Stolfi. Information contents of fracture lines. In *Proc. WSCG'2000 - 8th Intl. Conf. in Central Europe on Computer Graphics etc.*, volume 2, pages 389–395. Univ. of West Bohemia Press, February 2000.

[7] Mark Levoy. Scanning the fragments of the Forma Urbis Romae. WWW document available at `http: //www.graphics.stanford.edu/`, file `projects /mich/forma-urbis/forma-urbis.html`, May 1999.

[8] C. Menard and R. Sablatnig. On finding archaeological fragment assemblies using a bottom-up design. In *Proceedings of the 21st Workshop of the Austrian Association for Pattern Recognition*, pages 203–207, 1997.

[9] Farzin Mokhtarian. Silhouette-based isolated object recognition through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):539–544, 1995.

[10] Farzin Mokhtarian and Alan K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):789–805, 1992.

[11] Arthur R. Pope. Model based object recognition: A survey of recent research. Technical Report TR-94-04, Berkeley University, 1994.

[12] João Setubal and João Meidanis. *Introduction to Computational Molecular Biology*. PWS Publishing, 1997.

[13] Göktürk Üçoluk and I. Hakki Toroslu. Automatic reconstruction of broken 3-D surface objects. *Computers & Graphics*, 23(4):573–582, August 1999.

[14] Andrew P. Witkin. Scale-space filtering. In *Proceedings of IJCAI'83 - 8th International Joint Conference on Artificial Intelligence*, pages 1019–1021, 1983.

[15] Haim J. Wolfson. On curve matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(5):483–488, 1990.