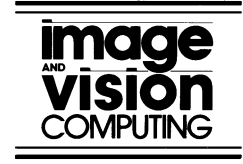




ELSEVIER

Image and Vision Computing 21 (2003) 401–412



www.elsevier.com/locate/imavis

On the automatic assemblage of arbitrary broken solid artefacts

Georgios Papaioannou^{*,1}, Evaggelia-Aggeliki Karabassi

Department of Informatics and Telecommunications, Faculty of Science, University of Athens, Panepistimioupolis Ilisia, Athens 157 84, Greece

Received 20 June 2001; received in revised form 23 November 2002; accepted 15 January 2003

Abstract

Presented here is a fast method that combines curve matching techniques with a surface matching algorithm to estimate the positioning and respective matching error for the joining of three-dimensional fragmented objects. Furthermore, this paper describes how multiple joints are evaluated and how the broken artefacts are clustered and transformed to form potential solutions of the assemblage problem.

© 2003 Elsevier Science B.V. All rights reserved.

Keywords: Curve matching; Depth buffer; Range maps; Optimisation methods; Surface matching

1. Introduction

The identification and joining of three-dimensional (3D) complementary objects, i.e. objects that can be combined to form a single solid object, is a problem that is receiving increasing attention by the computer science community. The problem is often encountered in archaeology and related fields and is closely associated with the assemblage of broken fragments in order to reconstruct the original artefacts.

Until recently, computer-aided assemblage of archaeological fragments was restricted to two dimensions, as in frescos [12] or relatively flat potsherds [8,15]. Advances in 3D scanning equipment have enabled introduction of 3D methodologies to address a wider range of artefacts.

In this paper we present a solution to the problem of object reconstruction from broken fragments of arbitrary 3D objects. As is the case in most real applications (especially in archaeology), no a priori knowledge is available regarding the number and the shape of the original objects that should be reconstructed from a collection of fragments, while usually a perfect match between the pieces is not possible, due to material deterioration and digitisation errors.

Fragmented objects are mostly encountered in the fields of archaeology and palaeontology, but sometimes also in

industrial inspection processes or medical applications. In order to be able to formulate a method applicable to such a wide range of objects, no assumptions concerning textural or structural features are made. In this paper, a fragment is regarded as a raw 3D model representing a broken solid piece of arbitrary shape with no other information. Fragments may have one or more broken surfaces, which they may share with more than one other broken pieces.

Common geometric features of the fragmented objects, assuming a moderate surface deterioration, are the irregularity of the broken surfaces and the sharp curvature transition from an intact surface to a broken one. These characteristics will be exploited in the method presented here to detect the broken surfaces and measure the matching between them. When a human tries to determine a possible match between two solids, the correspondence between the boundary lines of the surfaces is a major similarity criterion and a guide for gluing the pieces together. This important information is used in our method as a constraint to drastically limit the search space of the matching process and discard trivial and undesirable solutions.

1.1. Related work

In the two-dimensional (2D) case, where the complementary matching is reduced to a 'jigsaw puzzle', many solutions have been proposed which deal with the problem as matching of planar curve segments. Freeman [7] adopted the use of critical points to describe 2D shapes and measure shape features based on them. Ayache and Faugeras [1]

* Corresponding author. Tel.: +30-1-727-5107.

E-mail address: georgep@di.uoa.gr (G. Papaioannou).

¹ <http://www.di.uoa.gr/~georgep>

presented an algorithm for the matching of polygonal shapes under translation, rotation and scaling. Wolfson [23] proposed a method for the matching of smoothed polygonal approximations of 2D curves using string comparison on a signature based on the turning angle per sampled node.

Ucoluk and Toroslu [22] suggested a 3D curve matching approach for the joining of thin-walled fragments, based on string matching of the curvature and torsion scalar features of a discrete 3D curve. These features were earlier investigated by Kishon and Wolfson in Ref. [14]. The method is noise tolerant and allows for the matching of strings that contain incompatible segments. The authors propose this method for fragments that can be represented as closed 3D curves, such as potsherds, although no details are given regarding the discretisation, geometric matching of the curves and practical application examples.

Most matching algorithms that operate on 3D surface or volume data address the problem of object registration and are widely used in computer vision applications, such as object recognition, object classification, fusion and alignment of partial range data, as well as in molecular biology applications (e.g. protein docking). Registration and recognition methods that search for similarities over the entire objects, e.g. the Extended Gaussian Image [9], moment invariants [15,16], spherical harmonics [5], etc. cannot be applied in complementary matching so they will not be discussed further.

Methods that involve surface matching have some common features with the problem at hand, although they aim in the registration of identical curve and surface segments instead of the jointing of complementary (and possibly deteriorated) fragments.

One of the most popular algorithms for the registration of 3D objects of various types is the *iterative closest point* (ICP) algorithm, originally introduced by Besl and McKay [4]. Each iteration of ICP first finds the closest subset of a point set to another given point set based on geometric distance calculations. These point correspondences are used to calculate the least square rotation and translation transformations that register the two sets, using a quaternion-based approach [6,10]. The transformations are applied to one of the objects and the process is repeated until the objects are registered. It should be mentioned that as ICP converges to a local minimum, the method cannot guarantee the detection of the global distance minimum.

Barequet and Sharir [2] introduced a robust and noise tolerant method for the matching of point clouds representing the shell or the volume of partially identical objects. This method is based on the *geometric hashing* paradigm [11,24] and requires that user defined ‘footprints’ are calculated for or assigned to the evenly distributed points of the data sets. The method’s main disadvantages are that it requires uniform sampling of the objects while, according to the authors, it is difficult to be automated for arbitrary objects.

Papaioannou et al. [19] presented a method designed for 3D complementary matching of archaeological fragments, in the form of arbitrary polygonal surfaces. The method introduces a matching error between complementary surfaces that exploits the z-buffer algorithm. For each pair of surfaces the algorithm employs a stochastic search technique to minimise this matching error and derive the transformation that aligns the two fragments. The pairwise matching errors are used in an optimisation scheme to cluster the fragments into reconstructed objects. The method uses only surface information and does not take advantage of possible boundary curve similarities.

The work presented here extends the surface matching work of Papaioannou et al. [19] to incorporate curve matching ideas, similar to those of Ucoluk and Toroslu [22], providing a fast deterministic method for the complementary matching of fractured surfaces of solid objects under a boundary line similarity constraint.

1.2. Conventions and definitions

In the following text, all vectors are denoted by bold italic letters. The dot (\cdot) sign represents the dot product, while \times marks the cross product between two vectors. For an integer sequence $i = 1, \dots, n$, the \oplus sign represents the modulo n circular addition: $a \oplus b = (a + b - 1) \bmod(n) + 1$.

All geometric transformations are represented by their equivalent 4×4 transformation matrices, are denoted by capital bold letters and are applied to column vectors of 3D homogeneous coordinates: $\mathbf{b} = \mathbf{M} \cdot \mathbf{a}$, $\mathbf{a} = [a_x \ a_y \ a_z \ 1]^T$.

A matrix $\mathbf{R}_{v,\theta}$ represents the rotation around axis v by angle θ and \mathbf{T}_t is a translation by vector \mathbf{t} . \mathbf{S}_v is the anisotropic scaling transformation by v_x , v_y and v_z . We use a right-handed (CCW) coordinate system with vectors x , y and z being the axes of the global reference system. A matrix \mathbf{Z}_v represents the transformation that aligns a vector \mathbf{v} with axis z :

$$\mathbf{Z}_v = \mathbf{R}_{x,\theta} \mathbf{R}_{y,\varphi} \quad (1)$$

where $\varphi = -\text{atan}(v_x/v_z)$, $\theta = \text{atan}(v_y/\sqrt{v_x^2 + v_z^2})$. Note that $\mathbf{Z}_v^{-1} = \mathbf{R}_{y,-\varphi} \mathbf{R}_{x,-\theta}$.

The derivative of a function is represented by an accent. For instance, $r'(t)$ is the derivative of $r(t)$ and $r''(t)$ is its second derivative.

2. Method overview

The matching method operates on object surface meshes of arbitrary topology and it can be easily extended to work on volume crusts. As a preprocessing stage, the input meshes must be segmented into areas of adjacent nearly coplanar polygons, corresponding to crude *facets* or sides of the object and the potentially fractured facets must be identified. This procedure is described in Ref. [17], is briefly explained in Section 2.1 and need only be performed once,

after the fragment meshes are generated. At a second stage, fragments are processed in pairs, in order to define a transformation that matches their fractured facets and a *matching error* is calculated. This stage uses a complementary surface matching algorithm, in conjunction with facet boundary curve matching, whenever the later is applicable. Finally, an optimisation scheme is employed to arrange the fragment collection in a set of reconstructed objects, based on the pairwise matching errors, and the corresponding geometrical transformations are applied to arrange the fragments.

2.1. Detection of fractured sides

The surface segmentation of a fragment [17] is accomplished, using a simple region-growing algorithm. The process begins with an arbitrary polygon. Neighbouring polygons are classified to the same region if their normal vectors comply with a similarity criterion, otherwise a new region is formed. This criterion is either of a global nature, i.e. the new polygon's normal should not deviate from the average region normal by more than a predefined threshold, or a local one that includes the polygon in the current region if it is nearly coplanar with the adjacent polygons that have been already classified in the region.

During the region growing process, small surface regions may be created within larger ones. As it is desirable to partition the mesh into 'crude' facets, a cleanup stage eliminates small erroneous regions by iteratively assigning the polygons of small surface areas to large adjacent regions. A region is regarded as 'small' if it covers less than 5% of the entire mesh surface area.

Next, the algorithm proceeds by labeling as potential for matching those facets that exhibit higher coarseness. A facet's bumpiness can be estimated with an image-based bumpiness measure calculated on the depth map of the facet. This map is easy to obtain as it equals the contents of the z-buffer, after the facet triangles are rendered with the viewing direction parallel to the average facet normal. The bumpiness of a surface is associated with the rate of elevation variance and can be effectively estimated on the elevation map with an image filter, such as the Laplace image operator. This stage may also mark as broken, facets that are extensively engraved. However, this is not a problem as these facets are incompatible with any other and will therefore produce a high matching error during the error matching calculation. In any case, engraved surfaces may be distinguished from broken ones using pattern analysis, texture information or manual intervention.

2.2. Fragment matching

Given two fragment meshes Obj_1 , Obj_2 and two fractured facets $F_{1,m} \subset Obj_1$, $F_{2,n} \subset Obj_2$ we seek to calculate a complementary matching error $e_s(m,n)$ between the two

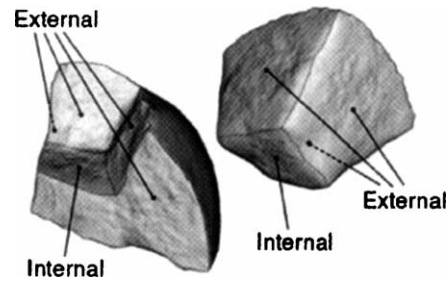


Fig. 1. The characterisation of a fragment with regard to which facet is under inspection.

fractured surfaces and define the rigid transformation $M_{m,n}$ that aligns Obj_1 with Obj_2 so that the complementary matching between $F_{1,m}$ and $F_{2,n}$ is optimal. The respective transformation that aligns Obj_2 with Obj_1 is simply: $M_{n,m} = M_{m,n}^{-1}$.

Depending on whether the two fragments are *external* or *internal*, two different surface matching approaches can be employed. We use the term *external* to characterise a fragment that has at least one facet that is both adjacent to the fractured facet and belongs to the surface of the original (intact) object. All other fragments are characterised as *internal*. If a fragment has multiple fractured facets, the fragment's characterisation depends on the fractured facet under examination, i.e. a fragment may be considered external when processing one of its fractured surfaces, and internal when processing another (Fig. 1). The extraction of the facet information that is required for the fragment characterisation is straightforward using a region adjacency graph of the segmented meshes.

In the most frequent case where both fragments are external, the fractured facet boundary information can be exploited to guide the search for complementary matching between the two fragments. However, curve matching alone is not sufficient and is used to constrain a surface matching algorithm. When at least one of the fragments is internal, boundary information is not reliable, so unconstrained surface matching is applied. Constrained matching is described in Section 3, while unconstrained matching is described in Section 4.

2.3. Object assemblage

Once pairwise matching errors have been calculated for all facet combinations, the fragments are clustered into groups that represent the reconstructed objects. Clustering is performed by selecting appropriate facet combinations and linking the fragments they belong to.

Assemblage rules are based on the work described in Ref. [19]. Selection of facet combinations should favour pairs that yield a smaller matching error. In order to avoid ambiguities, each fragment can be linked to as many other objects as the number of its fractured facets, while the link between two fragments should be unique.

Assemblage may be performed following two strategies, depending on the application:

- In cases where perfect matches are sought, a sorting of the available facet pairs is performed, with respect to the corresponding matching errors and combinations with smaller errors are selected first. As this selection proceeds, linked facets become unavailable as they cannot participate in new links and the number of possible new combinations is progressively diminished. This strategy leads to a rather small number of tight matches and is most appropriate when it is suspected that the fragment collection includes members that may not belong to any valid combination, as these members will be discarded in the process.
- If a large percentage of the fragments are expected to participate in valid combinations, an approach that minimises the sum of the matching errors of the individual combinations is adopted. This approach generates more fair solutions but diminishes the importance of perfect matches. More specifically, the set of fragment combinations that yields the smallest cumulative error is determined using exhaustive search. However, as exhaustive search execution time increases exponentially with respect to the number of fragments, a dynamic programming solution is preferable for large data sets. For this task, we use a genetic algorithm, as proposed in Ref. [18] and used in Ref. [19] for the same task, although other global optimisation methods could be adopted. The choice between an exhaustive search and dynamic programming depends on the desired execution time and the available processing speed.

The clustering procedure yields a graph whose nodes are the fragments of the collection and edges represent the connected facets. For a pair of connected facets of two fragments, namely $F_{i,m} \subset Obj_i$, $F_{j,n} \subset Obj_j$, the relative geometric transformations $\mathbf{M}_{m,n}$, $\mathbf{M}_{n,m}$ that correctly align the pieces have been calculated during the fragment matching stage and stored along with each fragment. In order to assemble the clustered pieces geometrically, these transformations are applied recursively to all pieces.

3. Boundary-constrained matching

The constrained search for a proper joint between two pieces Obj_1 , Obj_2 requires the extraction of the closed boundary curve of every surface facet candidate for matching. Then all correspondences between the boundaries of all pairs of broken facets $F_{1,m}$, $F_{2,n}$ are located and the fragments are aligned according to the appropriate transformation $\mathbf{M}_{m,n}$. All solutions that cause one surface to penetrate significantly into the other are discarded at this

stage. For the remaining solutions, if the fragments slightly intersect each other, the relative transformation $\mathbf{M}_{m,n}$ is corrected to separate them. For all acceptable solutions the corresponding surface matching error $e_s(m,n)$ is estimated. The configuration with the minimum error is considered the best pose for the facet pair $F_{1,m}$, $F_{2,n}$.

3.1. Boundary extraction

In order to achieve a boundary curve sampling independent of topology and surface representation, an image-based curve extraction procedure that uses the z-buffer is performed. This way it is possible to compare objects of different types (polygonal meshes, parametric surfaces, volume data) in a concise and unified manner even on fragment representations with topological errors (e.g. T-junctions or self-intersections).

For an object Obj_k , each surface facet $F_{k,m}$ has an average normal vector $\mathbf{n}_{ave}(F_{k,m})$ associated with it. In the case of a polygonal mesh with M_k polygons P_1, \dots, P_{M_k} , these vectors are:

$$\mathbf{n}_{ave}(F_{k,m}) = \left\| \sum_{j|P_j \in F_{k,m}} A_j \mathbf{n}_j \right\|^{-1} \cdot \sum_{j|P_j \in F_{k,m}} A_j \mathbf{n}_j \quad (2)$$

where A_j is the area of polygon P_j and \mathbf{n}_j is its normal vector. The discrete approximation of the facet boundary $H_{k,m} = \{ \mathbf{h}_{k,m}(1), \dots, \mathbf{h}_{k,m}(M_m) \}$, M_m being the number of points in $H_{k,m}$, is derived in the following manner.

First, facet $F_{k,m}$ is transformed in a way that $\mathbf{n}_{ave}(F_{k,m})$ coincides with the z axis and then it is rendered into an $N_b \times N_b$ z-buffer using an orthographic projection \mathbf{P}_{buf} . The image space representation of the facet is given by:

$$F_{k,m}^{(proj)} = \mathbf{P}_{buf} \mathbf{Z}_{\mathbf{n}_{ave}} \mathbf{T}_{-c} F_{k,m} \quad (3)$$

where $\mathbf{Z}_{\mathbf{n}_{ave}}$ is the alignment transformation of Eq. (1), c is the centroid of all vertices in $F_{k,m}$ and

$$\mathbf{P}_{buf} = \mathbf{T} \left(\frac{N_b}{2}, \frac{N_b}{2}, \frac{1}{2} \right) \mathbf{S} \left(\frac{N_b}{2R}, \frac{N_b}{2R}, \frac{-1}{2R} \right) \quad (4)$$

The normalising factor R is defined for each pair of fragments as the maximum radius of the two data sets.

The surface is rendered (simple scan-conversion in the case of polygonal meshes) and subsequently we extract the outer boundary of all non-background pixels stored in the depth buffer (Fig. 2) The depth buffer need not be too large. A 256×256 buffer is more than adequate considering that fine details are not very important for boundaries of damaged (fractured) surface regions. The extracted closed polyline $H_{k,m}^{(proj)}$ is transformed back to the standard coordinate system by the inverse transformation, to give $H_{k,m}$.

As the boundary line has been calculated on discrete data (the buffer cells), $H_{k,m}$ must be smoothed. Although

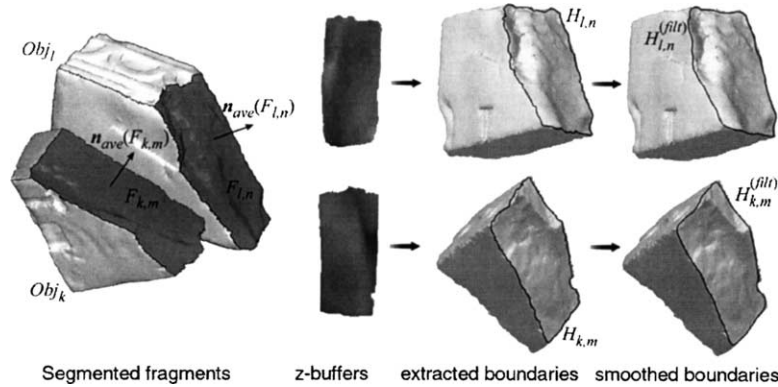


Fig. 2. The extraction of the fractured surface boundaries.

alternative smoothing techniques could be applied, satisfactory results were obtained by using a simple Gaussian filter f_G on the nodes $\mathbf{h}_{k,m}(i)$ (Fig. 2):

$$h_{k,m}^{(\text{filt})}(i) = \sum_{j=-w}^w \mathbf{h}_{k,m}(i-j) \cdot f_G(j), \quad f_G(i) = \frac{e^{-i^2/2\sigma^2}}{\sum_{j=-w}^w e^{-j^2/2\sigma^2}} \quad (5)$$

Filter width w depends on the buffer resolution. For a 256×256 z-buffer, a filter of width $w = 6$ performs well. w changes linearly with respect to the resolution. As nodes with a distance greater than 3σ from the central node have a negligible contribution when applying a Gaussian filter with standard deviation σ , the filter is designed using $\sigma = w/3$.

The above filter allows the subsampling of curve $H_{k,m}^{(\text{filt})}$, to obtain a new curve $B_{k,m} = \{\mathbf{b}_{k,m}(l), \dots, \mathbf{b}_{k,m}(N_m)\}$, N_m being the number of points in $B_{k,m}$. From the spatial frequency response of the filter and the Nyquist criterion, the maximum sampling period is $w/4$.

In order to compare the boundary lines of two fragments' facets, each curve $B_{k,m}$ is described by a signature $\mathbf{v}_{k,m}(i) = [k_{k,m}(s) \ \tau_{k,m}(s)]^T$, based on the discrete curvature $k(s)$ and torsion $\tau(s)$, where $s = s(i)$ is the arc length, as is the practice in Ref. [22]. $k = k(s)$ and $\tau = \tau(s)$ are estimated using discrete approximations of derivatives by average differences and are given by:

$$k = \|\mathbf{b}''\| \quad (6)$$

$$\tau = \frac{\mathbf{b}' \cdot (\mathbf{b}'' \times \mathbf{b}''')}{\|\mathbf{b}''\|^2} = \frac{[\mathbf{b}' \mathbf{b}'' \mathbf{b}''']}{k^2} \quad (7)$$

3.2. Boundary matching

The first stage of matching algorithm is the search for all matching segments between the boundary lines of two facets $F_{1,m}$, $F_{2,n}$ of two fragments Obj_1 , Obj_2 . This is addressed as a circular sub-string matching problem

between the signatures $\mathbf{v}_{1,m}$ and $\mathbf{v}_{2,n}$, respectively. The methodology adopted for this task is in many ways similar to the one proposed by Ucoluk and Toroslu [22] for the matching of thin objects represented as closed curves in 3D space. Although their method is of $O(N^2)$ complexity with respect to the number of samples per curve, the efficient sampling described in Section 3.1 guarantees small values on N .

The string matching is based on an $N_m \times N_n$ similarity matrix Λ , whose elements $\Lambda(i,j)$ hold the difference between signature node $\mathbf{v}_{1,m}(i)$ and $\mathbf{v}_{2,n}(j)$, expressed as the mean Euclidean distance of $\mathbf{v}_{1,m}(i)$ and $\mathbf{v}_{2,n}(j)$:

$$\Lambda(i,j) = \|\mathbf{v}_{1,m}(i) - \mathbf{v}_{2,n}(j)\| \quad (8)$$

Ucoluk and Toroslu construct such a matrix and detect similar signature segments as sequences of consecutive diagonal elements of $\Lambda : \{\Lambda(i,j), \Lambda(i \oplus 1, j \oplus 1), \dots, \Lambda(i \oplus L, j \oplus L)\}$, L being the length of detected segment. They also permit gaps to exist in similar segments and thus propose an algorithm for the combination of non-adjacent sequences to form longer strings.

Unlike their algorithm, we do not opt for permitting gaps to exist when locating similar signature sub-strings. Due to the differential nature of the signature attributes, the presence of dissimilar elements in a signature segment implies that there may exist substantial differences between the respective boundary segments.

Furthermore, in Ref. [22], the authors search for matching of non-overlapping closed-curves, each one describing an entire potsherd of negligible thickness. On the contrary, in our case, we require the overlap of the surface regions (fragment facets) that the boundary lines enclose. So, as all (directional) boundary lines have been extracted in a consistent manner, after aligning the average facet normal with the same axis (positive z -axis), we compare the strings mirrored, to reflect the fact that the facets are facing each other (Fig. 3). This amounts to a search for similar consecutive anti-diagonal elements of Λ (top-right to bottom-left with wrap-around): $\{\Lambda(i,j), \Lambda(i+1, j-1), \dots, \Lambda(i+L, j-L)\}$.

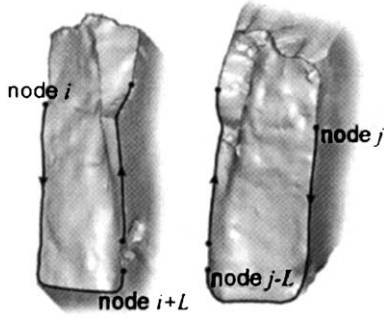


Fig. 3. Detection of matching boundary segments.

In our implementation $\Lambda(i, j)$ is measured as the mean Euclidean distance of $\mathbf{v}_{1,m}(i)$ and $\mathbf{v}_{2,n}(j)$ in the vicinity of the two nodes, using normalised torsion and curvature values:

$$\Lambda(i, j) = \frac{1}{3} \sum_{q=-1}^1 \|\mathbf{v}_{1,m}(i+q) - \mathbf{v}_{2,n}(j-q)\| \quad (9)$$

We require that the facets share edges at least 1/4 the arclength of the shortest boundary, which means that L should be roughly greater than $\min(N_m, N_n)/4$, but this depends on the application demands. As an ideal match between the boundary lines is rarely expected, we assume a similarity between two curve nodes if $\Lambda(i, j) < \text{tol}_\Lambda$, which has been set experimentally to $\text{tol}_\Lambda = 0.2$ for all available test cases.

After marking all similar signature segments, the detected substrings contained in larger ones are eliminated and the remaining curve segments are sorted in descending length order. If a total of N_{seg} pairs of segments are marked as similar, the next step is to find the rigid motion transformations $\mathbf{M}_{m,n}^{(i)}$, $i = 1, \dots, N_{\text{seg}}$ that align the segments of each pair i . This process will lead to N_{seg} transformations, corresponding to all possible valid boundary alignments, i.e. alignments that lead to a significant match between the two curves. These alignments will act as constraints in the next phase, where surface matching between the two facets will be attempted.

Each relative alignment transformation $\mathbf{M}_{m,n}^{(i)}$ between the points of the i -th segment pair can be calculated using a quaternion-based rigid motion estimation method, like the one suggested by Horn [10]. This technique performs well when the two boundary segments are almost identical; however in our experiments this was not usually the case and the method has not proved reliable for weathered fragments or under the presence of significant noise. For this reason, we have adopted an alternative, closed-form solution that operates on consecutive triads of corresponding points. This is described in Appendix A.

Each transformation $\mathbf{M}_{m,n}^{(i)}$ that aligns the m -th facet of the first fragment with the n -th facet of the second one, is associated with a matching error $e_b^{(i)}$ equal to the average distance between the aligned boundary segment points.

This error can be used to eliminate all incompatible relative poses of the fragments with regard to their facet boundaries. More precisely, if $e_{b,\min}$ and $e_{b,\max}$ are the minimum and maximum boundary matching errors, we discard all segment pairs with:

$$e_b^{(i)} > a_b \cdot e_{b,\min} + (1 - a_b) e_{b,\max}, \quad 0 < a_b \leq 1 \quad (10)$$

The choice of a_b is not critical as it only affects the overall speed of the method. We experimentally set a_b to 0.8.

3.3. Surface matching

The set of transformations $\mathbf{M}_{m,n}^{(i)}$, $i = 1, \dots, N_{\text{seg}}$ that align the facet boundary segments are not enough to determine a correct matching between two broken facets $F_{1,m}$, $F_{2,n}$ of the fragments Obj_1 and Obj_2 . Instead, a full surface comparison must be made and a surface matching error $e_s(m, n)$ estimated, using $\mathbf{M}_{m,n}^{(i)}$ as a constraint. First, we detect if the two surfaces penetrate each other. If the penetration depth is non-negligible $\mathbf{M}_{m,n}^{(i)}$ is discarded. In the opposite case, the fragments are separated and $e_s(m, n)$ is estimated.

The surface penetration and the error $e_s(m, n)$ are both related to the distance between corresponding points on the two facets. We can eliminate the computationally expensive registration of points and estimate the point-to-point distances directly if the fragment surfaces are sampled over a regular grid on a properly aligned reference plane (Fig. 4a). The distance sampling uses the z-buffers of the facets and is described next.

If the fragment facets under inspection are compatible and share at least one significant boundary segment, the alignment transformation $\mathbf{M}_{m,n}$ should reorient fragment Obj_1 so that $F_{1,m}$ faces $F_{2,n}$. This means that the projection of $F_{1,m}$ on $F_{2,n}$ and vice versa is maximised and the average normal vectors of the facets are roughly aligned: $\mathbf{n}_{\text{ave}}(F_{2,n}) \approx -\mathbf{n}_{\text{ave}}(F_{1,m})$. Therefore, if we pick a reference plane perpendicular to $\mathbf{n}_{\text{ave}}(F_{2,n})$, the point-to-point distances in the direction of $\mathbf{n}_{\text{ave}}(F_{2,n})$ equal the sum of distances of the corresponding points on Obj_1 and Obj_2 from the reference plane.

By aligning $\mathbf{n}_{\text{ave}}(F_{2,n})$ with the positive z axis, the reference plane becomes the xy plane, and the respective distances are directly derived from the z -buffers of the objects, if these are rendered into a rectangular image of resolution $N_b \times N_b$ using orthographic projection (Fig. 4b and c). Fragment Obj_2 is translated so that the centroid \mathbf{c} of facet $F_{2,n}$ lies on the centre of the orthographic viewing frustum, rotated to let $\mathbf{n}_{\text{ave}}(F_{2,n})$ become parallel to z , and orthographically projected using the projection matrix of Eq. (4):

$$Obj_2^{(\text{proj})} = \mathbf{P}_{\text{buf}} \mathbf{Z}_{\mathbf{n}_{\text{ave}}(F_{2,n})} \mathbf{T}_{-\mathbf{c}} Obj_2 \quad (11)$$

Fragment Obj_1 undergoes a similar transformation after being aligned to Obj_2 :

$$Obj_1^{(\text{proj})} = \mathbf{P}_{\text{buf}} \mathbf{S}_{(1,1,-1)} \mathbf{Z}_{\mathbf{n}_{\text{ave}}(F_{2,n})} \mathbf{T}_{-\mathbf{c}} \mathbf{M}_{m,n} Obj_1 \quad (12)$$

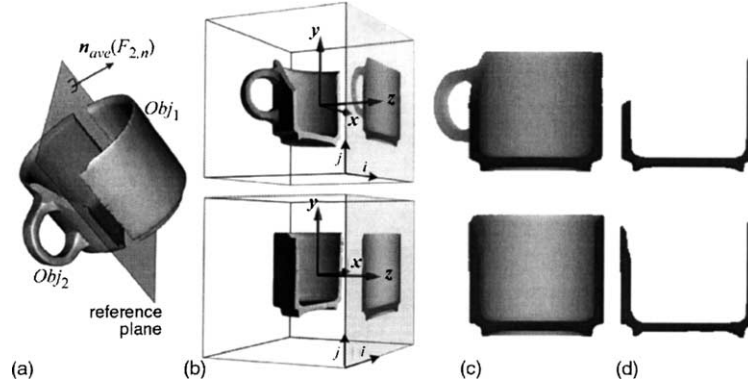


Fig. 4. Surface matching. (a) The relative positioning of the fragments. (b) Rendering of the fragments and acquisition of the z-buffers. (c) z-buffers of the entire fragments. They are used for the surface intersection detection. (d) The z-buffers of the current facets, which are used for the surface matching.

Note that the mirroring $\mathbf{S}_{(1,1,-1)}$ along the z axis is necessary as facet $F_{1,m}$ of Obj_1 faces the negative z axis.

Subsequently, both Obj_1 and Obj_2 are rendered separately into two z-buffers $D_1(i,j)$ and $D_2(i,j)$, $i, j = 1, \dots, N_b$. The elements of D_1 and D_2 are the normalised distances of the two fragments from plane xy and therefore $D_1(i,j) + D_2(i,j)$ is the distance of the surfaces of Obj_1 and Obj_2 measured at the (i, j) point of the regular grid.

The use of the depth maps eliminates the need for specific topology and regular or dense sampling of point clouds during the surface construction, thus making the method appropriate for arbitrary surfaces.

3.3.1. Surface intersection

As the contents of the z-buffers are normalised to the range $[0,1]$ ($0 = \text{near}$, $1 = \text{far}$), the surface penetration percentage at a grid location (i, j) is:

$$\text{penetration}(i,j) = 1 - D_1(i,j) - D_2(i,j) \quad (13)$$

The maximum penetration percentage $\text{penetration}_{\max}$ between the aligned fragments is simply the maximum value of $\text{penetration}(i,j)$. A relative positioning of two fragments Obj_1 and Obj_2 is discarded if:

$$\text{penetration}_{\max} > \text{tol}_{\text{pen}} \quad (14)$$

The threshold tol_{pen} should be small, even for severely damaged surfaces (2–5%). If the surface penetration is less than tol_{pen} , the transformation $\mathbf{M}_{m,n}$, that aligns Obj_1 with Obj_2 is corrected by a translation \mathbf{T}_{corr} along the average surface normal $\mathbf{n}_{\text{ave}}(F_{2,n})$:

$$\mathbf{t}_{\text{corr}} = \text{penetration}_{\max} \cdot R \cdot \mathbf{n}_{\text{ave}}(F_{2,n}) \quad (15)$$

where R is the coordinate normalisation factor of the orthographic projection as used in Eq. (4) and is used to scale the percentage $\text{penetration}_{\max}$ to a displacement in object space.

3.3.2. Surface matching error

If $\mathbf{M}_{m,n}$ produces a fragment positioning that passes the surface intersection test (no significant penetration), the two

corresponding fragment facets $F_{1,m}$ and $F_{2,n}$ are compared point by point and a surface matching error $e_s(m, n)$ is derived.

$e_s(m, n)$ depends on the facets' point-to-point distances, which are calculated in the same manner as in the case of the surface penetration. This time though, instead of rendering the entire fragments, only the two facets $F_{1,m}$ and $F_{2,n}$ are rendered into the z-buffers $D_1(i,j)$ and $D_2(i,j)$, $i, j = l, \dots, N_b$, reusing the transformations of Eqs. (11) and (12) (Fig. 4d).

The surface matching error may be estimated as the average point-to-point distance between the two facets. The minimum distance D_{\min} between the facets is subtracted, as it should not affect the match:

$$e(m, n) = \frac{1}{N_S} \sum_{(i,j) \in S} (D_1(i,j) + D_2(i,j)) - D_{\min} \quad (16)$$

where S is the set of depth buffer cells where both depth buffers have non-background values (i.e. $\neq 1$) and N_S is the number of elements in S .

The above formula presents the disadvantage that even small local surface distortions have a significant impact on the matching error. We adopt instead a matching error formula based on the surface derivatives), as proposed in Refs. [19,20]. These are easy to estimate over the already regularly sampled surfaces:

$$e_s(m, n) = \frac{1}{N_S} \sum_{(i,j) \in S} (|\Delta_x D_1(i,j) + \Delta_x D_2(i,j)| + |\Delta_y D_1(i,j) + \Delta_y D_2(i,j)|) \quad (17)$$

where $\Delta_x D(i,j)$ and $\Delta_y D(i,j)$ are the discrete approximation of the partial derivatives with regard to x and y directions. The matching error of Eq. (17) is noise tolerant and local distortions on the surfaces have a local effect.

If N_{S_1} and N_{S_2} is the number of non-background points for buffer D_1 and D_2 , respectively, we bias the error towards maximum surface overlap by multiplying Eq. (17) by

a weight factor:

$$\lambda_d = \frac{\min(N_{S_1}, N_{S_2})}{N_S} \quad (18)$$

After the calculation of $e_s(m, n)$ for all candidate matches $\mathbf{M}_{m,n}^{(i)}$, $i = 1, \dots, N_{\text{seg}}$ of two facets $F_{1,m}$, $F_{2,n}$, the relative positioning transformation $\mathbf{M}_{m,n}^{(i)}$ with the smallest matching error is considered as the best match $\mathbf{M}_{m,n}$ between them.

4. Unconstrained surface matching

In the case where at least one of the fragments Obj_1 , Obj_2 is characterised as internal with respect to the facet under examination, its boundary line cannot constrain the search for matching between the two pieces. An alternative matching method must be deployed which calculates the relative positioning transformation $\mathbf{M}_{m,n}$ directly from the surface information of facets $F_{1,m}$, $F_{2,n}$.

The ICP genre of registration algorithms would be a good choice for the pose estimation if not for the fact that they converge to local minima of the corresponding registration error. Apart from that, the phase of registration between closet points between large polygonal surfaces in each iteration is a tedious operation because it involves too many point-in-polygon containment tests. Banjemaa and Schmitt [3] propose a solution to decrease the registration time of ICP using multiple z-buffers over meshes segmented with the Gauss sphere normal vector classification criterion; however the problem of convergence to local minima remains.

The method proposed by Barequet and Sharir [2], which produces solutions that minimise the deviation between local features (footprints) of the two surfaces, can be applicable to this problem if one uses as a footprint the partial derivatives of an initial sampling of the z-buffers (see Section 3.3.2). Still, for fractured surfaces with deformations or missing matter between them, such an algorithm is prone to converge to solutions that match small portions of the fragment facets.

An approach is required that cannot be easily biased towards local similarities of the fractured surfaces and which produces a globally optimum match. We adopted here the stochastic global optimisation scheme used in Refs. [19,20], with the necessary modifications to reflect the current formulation of the problem.

The composite relative positioning transformation $\mathbf{M}_{m,n}$ defines a relative orientation and an offset between the fragments. If the natural representation of Euler angles is used for the orientation, this means that the matching error $e_s(m, n)$ must be optimised over 6 degrees of freedom (3 for the rotation and 3 for the translation). The fact that $e_s(m, n)$ does not depend on the space between the fractured surfaces, reduces the translation parameters to 2, if the translation is expressed as a shift of Obj_1 parallel to the reference plane xy . $\mathbf{M}_{m,n}$ can be

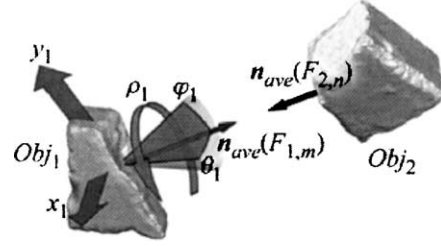


Fig. 5. The optimisation parameters of the unconstrained matching.

decomposed into:

$$\mathbf{M}_{m,n} = \mathbf{T}_{\mathbf{c}_2} \mathbf{Z}_{\mathbf{n}_{\text{ave}}(F_{2,n})}^{-1} \mathbf{M}_{\text{opt}} \mathbf{Z}_{-\mathbf{n}_{\text{ave}}(F_{1,m})} \mathbf{T}_{-\mathbf{c}_1} \quad (19)$$

where \mathbf{c}_1 and \mathbf{c}_2 are the centroids of facets $F_{1,m}$, and $F_{2,n}$ and \mathbf{M}_{opt} defines the transformation matrix to be optimised. The above representation of $\mathbf{M}_{m,n}$ allows all rotations and translations to be performed with regard to the global coordinate system, which is more intuitive and facilitates the calculation of the matching error.

If \mathbf{M}_{opt} is expressed as:

$$\mathbf{M}_{\text{opt}} = \mathbf{T}_{(x_1, x_2, 0)} \mathbf{R}_{x, \theta_1} \mathbf{R}_{y, \varphi_1} \mathbf{R}_{z, \rho_1} \quad (20)$$

we get a set of five parameters $(\theta_1, \varphi_1, \rho_1, x_1, y_1)$, 2 of which have a very limited range (Fig. 5). More specifically, fragment Obj_1 can perform a full circle around z (ρ_1) and slide along plane xy within the extents of the z-buffer. The rotations around x and y axes are limited though, because they reflect the expected deviation of the facet normal vectors. A range of ± 10 degrees is more than enough for angles θ_1, φ_1 .

The optimisation method implemented in our tests was a *simulated annealing* variant [13,20,21], but alternative global optimisation methods can be used as well.

5. The assemblage algorithm

To summarise the stages described in Sections 2–, the automatic assemblage method can be described in the following steps:

Segmentation

For all fragments Obj_k , $k = 1, \dots, N_{\text{obj}}$:

Step 1. Segment mesh Obj_k into adjacent facets.

Step 2. Detect the possibly fractured facets $F_{k,m}$,

$$m = 1, \dots, N_{k, \text{facets}}.$$

Matching

For all fragment pairs (Obj_k, Obj_l) , $k, l = 1, \dots, N_{\text{obj}}$, $k \neq l$:

For all fractured facet pairs $(F_{k,m}, F_{l,n})$,

$$m = 1, \dots, N_{k, \text{facets}}, n = 1, \dots, N_{l, \text{facets}}:$$

If $(Obj_k$ is internal with respect to $F_{k,m}$) OR $(Obj_l$ is internal with respect to $F_{l,n})$:

Estimate the optimised relative positioning transformation $\mathbf{M}_{m,n}$ between $F_{k,m}$, $F_{l,n}$, using unconstrained matching.

Else (constrained matching):

- Step 1. Extract the image-based boundary lines of $F_{k,m}$ and $F_{l,n}$.
- Step 2. Filter and resample the boundary lines to obtain curves $B_{k,m}$ and $B_{l,n}$.
- Step 3. Calculate the signatures $\mathbf{v}_{k,m}$ and $\mathbf{v}_{l,n}$ from curves $B_{k,m}$ and $B_{l,n}$.
- Step 4. Detect all N_{seg} pairs of similar segments of $B_{k,m}$ and $B_{l,n}$, based on $\mathbf{v}_{k,m}$, $\mathbf{v}_{l,n}$.
- Step 5. Find the transformations $\mathbf{M}_{m,n}^{(i)}$, $i = 1, \dots, N_{\text{seg}}$ that align the segments of each segment pair i .
- Step 6. Discard $\mathbf{M}_{m,n}^{(i)}$ that lead to large error in segment matching.
- Step 7. For all remaining $\mathbf{M}_{m,n}^{(i)}$ estimate the surface penetration between Obj_k and Obj_l .
- Step 8. Discard all $\mathbf{M}_{m,n}^{(i)}$ that lead to significant object intersection.
- Step 9. For all remaining $\mathbf{M}_{m,n}^{(i)}$ calculate the surface matching error $e_s(m, n)$.
- Step 10. Set as optimal transformation $\mathbf{M}_{m,n}$ the one with minimum $e_s(m, n)$.

Assemblage

- Step 1. Optimise the fragment facet combinations based on the calculated surface matching errors.
- Step 2. Geometrically arrange the fragments to form the final objects.

6. Case studies

The method has been tested with a variety of objects. Fragments were first digitised and then decimated, as very fine detail is superfluous and is also bound to have the impact of noise in the matching procedure. The digitisation process was carried out with a touch probe, which facilitated the generation of point clouds of varying density, depending on local surface irregularity and fragment size (sampling resolution between approx. 0.7 mm and 2 cm). Commercial software was used for the mesh generation from the point clouds, the model cleanup and polygon reduction of the resulting surfaces.

The average preprocessing time for facet segmentation and characterisation per facet was 0.14 s, for an average number of triangles per facet equal to 1870. The average curve extraction time per facet was 0.06 s. The pairwise facet matching time was 2.2 s per facet pair. The automatic assemblage stage, consisting of only a combination optimisation procedure and transformation composition steps, is performed in far less time than the fragment matching and was of an order of a few ms.

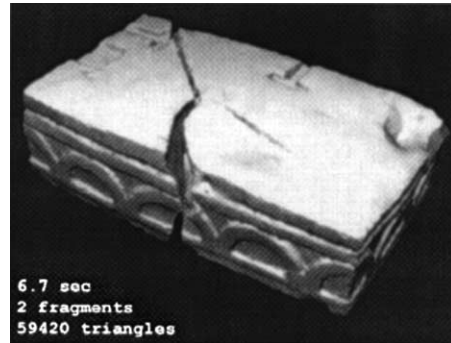


Fig. 6. Reassembly of an ornate plaster slab.

All tests discussed here were performed using constrained matching and show that the method performs well for a variety of objects. Some of the most characteristic cases are shown in Figs. 6–9. Each example is accompanied by the total number of triangles of all fragments under examination, the number of fragments and the total processing time required for preprocessing, curve extraction, pairwise matching error calculations and assemblage. The tests were performed on a Pentium III/450 MHz based PC.

Fig. 6 shows the successful assemblage of a broken ornate plaster block. The block was split into two large fragments, when dropped to hard ground from about 1.5 m. The fracture interface was significantly damaged as many smaller fragments came loose from both sides. In Fig. 7, a small collection of 10 fragments of 4 identical rectangular blocks was reassembled. The same data set was used to test unconstrained matching in Ref. [19] and the result presented several ambiguities, as all fragmented facets are very smooth and surface matching error was similar for different combinations. Most ambiguities were resolved when using the constrained matching presented here, as the curve matching guided effectively the method to produce the correct result. The only problem presented was in the case of the bottom right block: As the fragmented facets are rounded and give little joint area support, both boundary curves and surfaces match

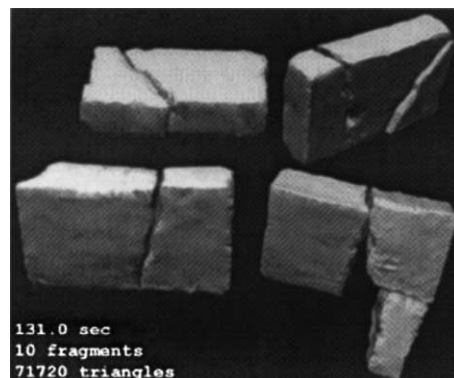


Fig. 7. A multiple part assemblage example.

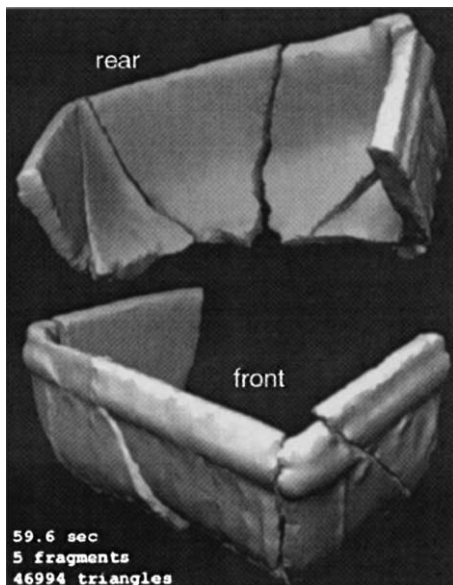


Fig. 8. Partial reconstruction of a broken pot.

only over small areas and they can not be combined (even by a human) without a priori knowledge of the desired shape of the reconstructed object.

Fig. 8 shows the partial reconstruction of a rectangular clay pot. Five pieces were digitised in rather low resolution (average 8.7 samples/cm²) and used for the reassembling. The final case, in Fig. 9, involves two broken cups reconstructed by assembling a total of five fragments. A few minor fragments were discarded before digitisation.

In our collection of fragments, there was no piece that could not be handled by boundary line constrained matching and therefore we could only do a comparative study with unconstrained matching using the same pieces. By comparing the results of the constrained surface matching with previous results on unconstrained matching [18,19] as well as current test cases, the later proves less efficient and more computationally intensive, as expected. The relative pose estimation time in unconstrained matching is about four times greater than the constrained one. Therefore, it is apparent that boundary constrained matching is the method of choice wherever applicable and

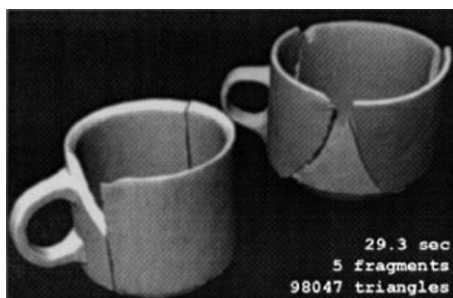


Fig. 9. Reassemblage of two cups from five fragments.

unconstrained matching should be used only when boundary lines cannot be successfully extracted or do not provide concrete evidence of the relation between two fragments.

7. Conclusion

This paper presented a complete methodology for the full or partial assemblage of arbitrary 3D objects from parts, using as input only the surface representation of the fragments. The method exploits and extends previous research results to combine curve and surface matching algorithms in order to address the assemblage problem in a unified manner for all types of objects.

The surface segmentation, curve extraction and surface matching procedures use the z-buffer algorithm so the method can operate on a variety of object data representations and is independent of the initial surface sampling and complexity. Computing the various metrics in the z-buffer image space has the additional advantage that the method is inherently multiresolutional. For presentation simplicity, we have not described a progressive surface error measurement scheme but this is quite straightforward.

If additional information about the fragments to be assembled is available, e.g. material properties, it can be effectively incorporated in the form of constraints on the combination optimisation of the final assemblage stage.

Appendix A. Alignment of two segments

In order to find the best transformation that aligns two curve segments $Seg_1 = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$ and $Seg_2 = \{\mathbf{q}_1, \dots, \mathbf{q}_N\}$, we process the points of the two segments in triads: $\{\mathbf{p}_i, \mathbf{p}_{i+1}, \mathbf{p}_{i+2}\}$, $\{\mathbf{q}_i, \mathbf{q}_{i+1}, \mathbf{q}_{i+2}\}$, derive a set of $N - 2$ transformations \mathbf{A}_i , $i = 1, \dots, N - 2$, and keep the one that minimises the average distance between the two entire point sets Seg_1 and Seg_2 . In the following it is assumed that point \mathbf{p}_i , $i = 1, \dots, N$, of Seg_1 corresponds to \mathbf{q}_i of Seg_2 (Fig. 10a).

As a first step we translate both triads so that both \mathbf{p}_i and \mathbf{q}_i coincide with the coordinate origin (Fig. 10b). These translations are $\mathbf{T}_{-\mathbf{p}_i}$ and $\mathbf{T}_{-\mathbf{q}_i}$, respectively.

In order to match \mathbf{p}_{i+1} with \mathbf{q}_{i+1} , we transform the triads so that the unit vectors $\mathbf{d}(\mathbf{p}_{i+1}) = (\mathbf{p}_{i+1} - \mathbf{p}_i) / \|\mathbf{p}_{i+1} - \mathbf{p}_i\|$ and $\mathbf{d}(\mathbf{q}_{i+1}) = (\mathbf{q}_{i+1} - \mathbf{q}_i) / \|\mathbf{q}_{i+1} - \mathbf{q}_i\|$ are aligned with the common reference axis. If axis z is used, the corresponding transformations are $\mathbf{Z}_{\mathbf{d}(\mathbf{p}_{i+1})}$ and $\mathbf{Z}_{\mathbf{d}(\mathbf{q}_{i+1})}$, as in Eq. (1) (Fig. 10c).

The above transformations (translation and alignment) lead to two new triads $\{\tilde{\mathbf{p}}_i, \tilde{\mathbf{p}}_{i+1}, \tilde{\mathbf{p}}_{i+2}\}$ and $\{\tilde{\mathbf{q}}_i, \tilde{\mathbf{q}}_{i+1}, \tilde{\mathbf{q}}_{i+2}\}$.

Once the first two points are aligned we try to transform $\tilde{\mathbf{p}}_{i+2}$ as to coincide with $\tilde{\mathbf{q}}_{i+2}$. This is accomplished by aligning the two planes that are defined by the transformed

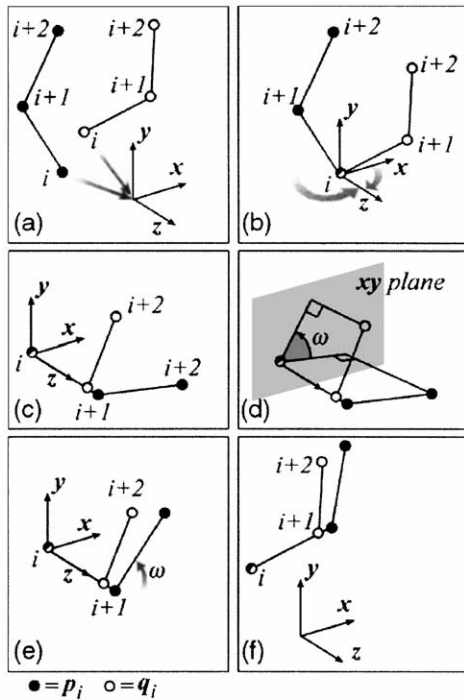


Fig. 10. The procedure steps of the registration of point triads.

point triads. Let ω be the angle between the two planes (Fig. 10d). This angle corresponds to the angle between the projections of $\mathbf{d}(\tilde{\mathbf{p}}_{i+2})$ and $\mathbf{d}(\tilde{\mathbf{q}}_{i+2})$ on the xy plane:

$$\omega = \omega_2 - \omega_1,$$

where

$$\omega_1 = \text{atan}\left(\frac{\mathbf{d}_y(\tilde{\mathbf{p}}_{i+2})}{\mathbf{d}_x(\tilde{\mathbf{p}}_{i+2})}\right)$$

and

$$\omega_2 = \text{atan}\left(\frac{\mathbf{d}_y(\tilde{\mathbf{q}}_{i+2})}{\mathbf{d}_x(\tilde{\mathbf{q}}_{i+2})}\right).$$

As a result, the relative transformation that aligns triad $\{\mathbf{p}_i, \mathbf{p}_{i+1}, \mathbf{p}_{i+2}\}$ with triad $\{\mathbf{q}_i, \mathbf{q}_{i+1}, \mathbf{q}_{i+2}\}$ is (Fig. 10e and f):

$$\mathbf{A}_i = \mathbf{T}_{\mathbf{q}_i} \mathbf{Z}_{\mathbf{d}(\mathbf{q}_{i+1})}^{-1} \mathbf{R}_{z,\omega} \mathbf{Z}_{\mathbf{d}(\mathbf{p}_{i+1})} \mathbf{T}_{-\mathbf{p}_i}$$

Note that this alignment transformation is defined even if triad points are collinear. If the two triads do not perfectly match, \mathbf{A}_i still forces the first points to coincide, aligns the first edge and ensures that the two triads are coplanar.

For each \mathbf{A}_i is computed, we calculate the average segment distance:

$$e_i = \frac{1}{N} \sum_{j=1}^N \|\mathbf{q}_j - \mathbf{A}_i \mathbf{p}_j\|$$

The final transformation that is used to align the two segments, is the one corresponding to the minimum value of e_i .

References

- [1] N. Ayache, O.D. Faugeras, HYPER: a new approach for the recognition and positioning of two-dimensional objects, IEEE Transactions of PAMI 8 (1) (1986) 44–54.
- [2] G. Barequet, M. Sharir, Partial surface and volume matching in three-dimensions, IEEE Transactions of PAMI 19 (9) (1997) 929–948.
- [3] R. Benjemaa, F. Schmitt, Fast global registration of 3D sampled surfaces using a multi-z-buffer technique, Image and Vision Computing 17 (1999) 113–123.
- [4] P. Besl, N. McKay, A method for registration of 3-D shapes, IEEE Transactions of PAMI 14 (2) (1992) 239–256.
- [5] G. Burel, H. Henocq, Determination of the orientation of 3-D objects using spherical harmonics, Graphical Models and Image Processing 57 (5) (1995) 400–408.
- [6] O.D. Faugeras, M. Hebert, A 3-D recognition and positioning algorithm using geometrical matching between primitive surfaces, Proceedings of the International Joint Conference on Artificial Intelligence'83, Morgan-Kaufmann, Vancouver, 1983, pp. 996–1002.
- [7] H. Freeman, Shape description via the use of critical points, Pattern Recognition 10 (1978) 159–166.
- [8] K. Hori, M. Imai, T. Ogasawara, Joint detection for potsherds of broken earthenware, Proceedings of the CVPR'99, IEEE Computer Society, Fort Collins, CO, 1999, pp. 440–445.
- [9] B.K.P. Horn, Extended Gaussian images, Proceedings of the IEEE 72 (1984) 1656–1678.
- [10] B.K.P. Horn, Closed-form solution of absolute orientation using unit quaternions, Journal of the Optical Society of America A 4 (1987) 629–642.
- [11] A. Kalvin, E. Schonberg, J. Schwartz, M. Sharir, Two-dimensional model-based boundary matching using footprints, International Journal of Robotics Research 5 (4) (1986) 38–55.
- [12] A.D. Kalvin, A. Remy, L.J. Castillo, K. Morla, E. Nolasco, J. Prado, V. Fernandez, R. Franco, G. Wiese, Computer-aided reconstruction of a pre-Inca temple ceiling in Peru, Proceedings of the Computer Applications in Archaeology (CAA 97), University of Birmingham, UK, 1997.
- [13] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
- [14] E. Kishon, H. Wolfson, 3-D curve matching, Proceedings of the AAAI Workshop on Spatial Reasoning and Multi-sensor Fusion (1987) 250–261.
- [15] H.C.G. Leitao, J. Stolfi, Information contents of fracture lines, Proceedings of the Eighth International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media (WSCG'2000), University of West Bohemia Press, Plzen, Czech Republic, 2000, pp. 389–395.
- [16] C.H. Lo, H.S. Don, 3-D moment forms: their construction and application to object identification and positioning, IEEE Transactions of PAMI 11 (10) (1989) 1053–1064.
- [17] G. Papaioannou, E.A. Karabassi, T. Theoharis, Segmentation and surface characterization of arbitrary 3D meshes for object reconstruction and recognition, Proceedings of the ICPR'2000, IEEE Computer Society, Barcelona, 2000, pp. 734–737.
- [18] G. Papaioannou, E.A. Karabassi, T. Theoharis, Automatic reconstruction of archaeological finds—a graphics approach, Proceedings of the International Conference on Computer Graphics and Artificial Intelligence (3IA 2000), Limoges, France (2000) 117–125.
- [19] G. Papaioannou, E.A. Karabassi, T. Theoharis, Virtual archaeologist: assembling the past, IEEE Computer Graphics and Applications 21 (2) (2001) 53–59.
- [20] G. Papaioannou, E.A. Karabassi, T. Theoharis, Reconstruction of

- three-dimensional objects through matching of their parts, *IEEE Transactions of PAMI* 24 (1) (2002) 114–124.
- [21] P. Siarry, G. Berthiau, F. Durbin, J. Haussy, Enhanced simulated annealing for globally minimizing functions of many-continuous variables, *ACM Transactions of Mathematical Software* 23 (2) (1997) 209–228.
- [22] G. Ucoluk, I.H. Toroslu, Automatic reconstruction of 3-D surface objects, *Computers and Graphics* 23 (1999) 573–582.
- [23] H.J. Wolfson, On curve matching, *IEEE Transactions of PAMI* 12 (5) (1990) 483–489.
- [24] H.J. Wolfson, I. Rigoutsos, Geometric hashing: an overview, *IEEE Computational Science and Engineering* 4 (4) (1997) 10–21.