Bayesian Assembly of 3D Axially Symmetric Shapes from Fragments

Andrew R. Willis, David B. Cooper
Brown University, Division of Engineering, Providence, RI 02912, USA

Abstract

We present a complete system for the purpose of automatically assembling 3D pots given 3D measurements of their fragments commonly called sherds. A Bayesian approach is formulated which, at present, models the data given a set of sherd geometric parameters. Dense sherd measurement data is obtained by scanning the outside surface of each sherd with a laser scanner. Mathematical models, specified by a set of geometric parameters, represent the sherd outer surface and break curves on the outer surface (where two sherds have broken apart). Optimal alignment of assemblies of sherds, called configurations, is implemented as maximum likelihood estimation (MLE) of the surface and break curve parameters given the measured sherd data for all sherds in a configuration. The assembly process starts with a fast clustering scheme which approximates the MLE solution for all sherd pairs, i.e., configurations of size 2, using a subspace of the geometric parameters, i.e., the sherd break curves. More accurate MLE values based on all parameters, i.e., sherd alignments, are computed when sherd pairs are merged with other sherd configurations. Merges take place in order of constant probability starting at the most probable configuration. This method is robust to missing sherds or groups of sherds which contain sherds from more than one pot. The system represents at least three significant advances over previous 3D puzzle solving approaches : (1) a Bayesian framework which allows for easily combining diverse types of information extracted from each sherd, (2) a search which reduces comparisons on unlikely configurations, and (3) a robust computationally reasonable method for aligning break curves and sherd outer surfaces simultaneously. In addition, a number of insights are given which have not previously been discussed and significantly reduce computation. Methods proposed for (1),(2), and (3) represent important contributions to the field of puzzle assembly, 3D geometry learning, and dataset alignment and are critical to making 3D puzzle solutions tractable to compute. Results are presented which include assembling a 13 sherd pot where only an incomplete set of 10 sherds is available.

Keywords: automatic 3D puzzle assembly, 3D structure from unorganized 3D data, 3D alignment, geometric learning, perceptual grouping, hierarchical clustering.

1 Introduction

Our goal is to estimate 3D free-form surfaces from dense noisy 3D measurements of small pieces in a pile. We assume that collections of some of these pieces describe the unknown shapes which we seek to estimate. For each shape, there is a global constraint, namely, the shape is axially symmetric, i.e., symmetric about an axis, a.k.a. a circularly symmetric generalized cylinder. The radius function, i.e., cross section function, may be multivalued over intervals along the axis. Typically, a pile may contain 50-200 small pieces, and 20-40 may belong to a shape. Some of the small pieces may belong to no shape, and are therefore just clutter. The challenge here is that the pieces are small, hence how to estimate an accurate axis/profile-curve for a piece is not obvious and may even be not possible, the break-curves along which the surface breaks into pairs of pieces may be chipped and eroded, and the search space for assembling these pieces is enormous, i.e., all pairs of pieces must be checked for matching and alignment in all possible relative positions, and more generally, this must be done for groups of pieces using all available geometric information. This problem arises in the estimation of virtual pots from sherds at archaeological sites, but the ideas are central to the more general problem of inferring 3D geometry from measurement of fragments in the presence of clutter. This problem is a heretofore unsolved problem, though various theoretical approaches to the problem solution have been offered. Our software has automatically assembled a pot of 13 sherds from the 10 that could be scanned. In this paper, we formulate and implement a new solution, discuss the key ideas necessary for the success of the approach, and demonstrate a solution using a pot that has been broken into 13 sherds, thus providing ground truth.

A fundamental open problem in computer vision is matching objects based on their shapes. Shape-matching algorithms have a wide variety of applications: one important application is the automatic assembly of puzzles from their pieces. Early research on the subject includes works such as [8, 18, 4] and more recently [5]. Here puzzle pieces are commercially produced jigsaw puzzle pieces which simplifies the problem since jigsaw pieces have similar sizes, easily identifiable areas which need to be matched, and a mostly regular outline, i.e., the piece contour is differen-

from the Proceedings of CVPR (Conference on Computer Vision and Pattern Recognition) Vol I, June 2004

tiable with a few exceptions at corner pieces which require special treatment. Solving more difficult puzzles based on real-world data remains an active area of research today with recent results published in [11, 5, 7], which attempt to improve the algorithmic speed and remove these restrictions. Yet, examples of automatic puzzle solving in 3D are limited to [9, 7, 12]. In [7, 12], the authors deal with fragments of arbitrary shape and propose to assemble them elegantly by matching the fragments break curves, i.e., the curves on the pot surface along which the sherds break apart (Fig. 2). Both methods match curves based on their curvature and torsion signatures which, unfortunately, is prone to instability when the data is noisy. In [9], the authors match broken 3D shapes by matching of their fractured surfaces, i.e., the surface through the pot wall at which the pot breaks, using simulated annealing. Their approach requires both of the fractured surfaces to be parametrizable with respect to a common plane, i.e., for aligning 2 fractured surfaces about the xy-plane the two surfaces may be represented as $f_1(x, y) = z_1$ and $f_2(x, y) = z_2$. For all three methods, no results are provided for 3D fragments with more than 2 pieces. Specifically, no results for assembly of more than two 3D sherds are given in [7], a single pairwise match is shown in [12], and several pairwise matches are shown in [9].

We propose several improvements with respect to past 3D puzzle solving approaches : 1) a Bayesian framework which allows for easily combining diverse types of information extracted from each fragment, 2) an approximate uniform cost search which reduces comparisons on false positives, i.e., incorrect matches of high probability, by always using the most probable match available, and 3) a robust method for aligning break curves and sherd outer surfaces simultaneously. In addition, a number of insights are given which have not previously been discussed and significantly reduce computation. A working implementation of the Bayesian framework introduced in [17, 15, 16] and further developed in [14] is presented for the purpose of assembling pots from 3D measurements of their fragments which we refer to as *sherds*.

The big hurdles that had to be overcome are that sherds are often small so that little surface shape is discernable from the surface measurements, break curves may not be sharp due to erosion and chipping, a break curve shared by a pair of sherds may be short and not have distinctive shape, and the search space for matching pairs and groups of sherds is huge.

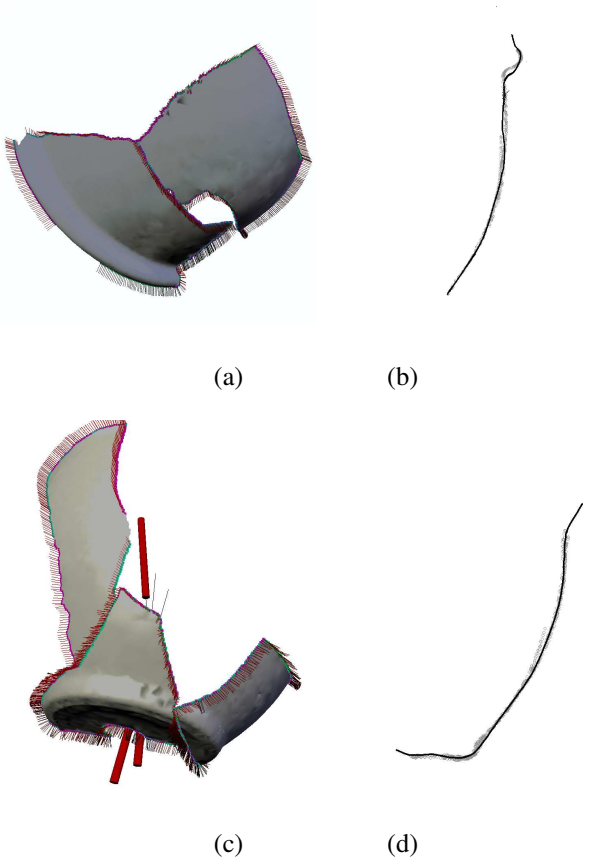

Figure 1: Incorrect matches (configurations) of high probability : Two configurations are shown which are incorrect. However the break-curve matches shown in (a) and (c) match well and the corresponding axis/profile-curves in (b) and (d) indicate that these are reasonable pot configurations. These are among the top 5 most probable pairs, e.g., for (a) and triplets, e.g., for (c), created by our assembly system. (b) and (d) are the profile curves for configurations (a) and (c), respectively. The shown point scattered about the profile curve are surface measured points in the direction perpendicular to the estimated axis for the configuration. Since they lie very close to the estimated profile curve for the configuration, the configuration, though incorrect, satisfies the requirement of being an axially symmetric surface.

2 Model Parameters

For the purpose of this paper, we have focused on a subset of the geometric information that can be used. It consists of the outer-surface break curves, break curve *vertices* at sherd junctions (Fig. (2)), axis/profile curve for the entire pot and portions for individual sherds, and Euclidean transformations that take each sherd from its data-measurement position to its aligned position in a configuration.

Our goal is to estimate the global pot parameters in Table (1) by hypothesizing matches, i.e., transformations, between sherd datasets. The most probable matched set of sherd data given the sherd transformation values and the geometric parameters is considered the most likely pot.

Symbol	Significance
$\mathbf{l}$	the vessel axis
$\boldsymbol{\alpha}$	the axially symmetric surface of the entire vessel (*i.e.*, profile-curve, $\alpha(z)$, with respect to z-axis)
$\boldsymbol{\beta}$	break-curves for entire vessel
$\mathbf{T}_i$	a 3D transformation which takes sherd i into its aligned position in the vessel

Table 1: Basic geometric parameters.

The pot break-curve parameters, $\boldsymbol{\beta}$, are the locations on the pot surface along which the pot breaks. These locations include *vertices*—locations of Y and T junctions (see Fig. (2)). Note, Y-junctions are points which are high-curvature points on sherd boundaries. T-junctions are points which are high-curvature points on two of the three sherd boundaries that meet at a T-junction. We also refer to these high-curvature points on sherd boundaries as *vertices.* The additional points that constitute our representation for the break curves is a sequence of K points for each curve segment starting on a vertex. The points in such as sequence occur at successive intervals of fixed length from the vertex (see Fig. (3)). We call such a sequence of K points a *break-point segment*. These points along with a surface normal at each point constitute $\boldsymbol{\beta}$—our parameterization of the break-curves for the pot. Hence, break-point segment v is written $\boldsymbol{\beta}_v = ((\mathbf{p}_1, \mathbf{p}_2, \ldots \mathbf{p}_K), (\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_K))$ where $\mathbf{p}_k$ denotes the k^{th} 3D point and $\mathbf{n}_k$ denotes the k^{th} 3D normal for $\boldsymbol{\beta}_v$. The group of all break-curve parameters is $\boldsymbol{\beta} = \cup_{v=1} \boldsymbol{\beta}_v$.

Each break-point segment corresponds to a location where the break-curve intersects a sphere of radius rR centered at the pot vertex where $r = [0, 1, 2, \ldots, K-1]$. When we measure sherd break-curve data, noisy estimates of these points are extracted using a set of hypothesized vertex points (see § 3). Figure (3) illustrates two break-point segments for a vertex where $K = 4$.

In [17] fast and robust methods are provided for the difficult problem of estimating $(\mathbf{l}, \boldsymbol{\alpha})$, the axis/profile-curve pairs from small patches of noisy 3D data measurements of axially-symmetric surfaces which is the case for archaeological pot fragment data. We have parameterized the pot axis of symmetry using the standard parametric equation of a 3D line as shown in (1).

$$\begin{aligned} x &= m_x z + b_x \,, \\ y &= m_y z + b_y \,. \end{aligned} \tag{1}$$

Hence $\mathbf{l} = (b_x, b_y, m_x, m_y)$ and consists of the pair (m_x, m_y), specifying the slope of the line when it is projected onto the xz-plane and the yz-plane, respectively, and the pair (b_x, b_y), specifying where the line intercepts the xy-plane at $z = 0$.

The profile curve $\alpha(r, z)$, with respect to the sherd axis $\mathbf{l}$, defines a *3D axially symmetric algebraic surface* with axis $\mathbf{l}$. The surface parameters are the coefficients of the algebraic profile curve in (2) and the axis of symmetry in (1). The form of the 2D algebraic profile curve of degree d is (2).

$$\alpha(r, z) \;\; = \sum_{0 \le j+k \le d;\; j,k \ge 0} \alpha_{jk} r^j z^k = 0 \tag{2}$$

Hence $\boldsymbol{\alpha} = (\cup_{j,k} \alpha_{jk})$ is the vector of coefficients for the implicit polynomial curve of degree d. Note that for our experiments, $d = 6$.

We assume that each sherd undergoes an arbitrary rigid Euclidean transformation which moves the sherd to it's measurement position. Hence, for the i^{th} sherd we must estimate the transformation, $\mathbf{T}_i$, which places the sherd into its aligned position in the pot. We parametrize a 3D transformation with 6 parameters consisting of 2 parts : (1) a 3D translation vector $\mathbf{t}$, and (2) a 3D rotation $\mathbf{R}$. The 3x3 rigid rotation matrix $\mathbf{R}$ is represented using the so called axis-angle parameters which describe rotation in terms of a rotation angle ψ about a 3D unit vector $\mathbf{n_R}$. Hence our rotation is the 3D vector $(\psi \mathbf{n_R})$ and we refer to the equivalent 3x3 rotation matrix as $\mathbf{R}$. The 3D transformation parameters are $\mathbf{T} = (\mathbf{t}, \psi \mathbf{n_R})$ (see Appendix A of [10] for additional details on this parameterization). Further, we denote transformations of sherd datasets as $\mathbf{T}(\mathcal{D})$ for surface point/normal data and $\mathbf{T}(\mathcal{B})$ for break-curve segment point/normal data. In this notation it is assumed that $\mathbf{T}(\mathcal{D})$ indicates that a single transformation, $\mathbf{T}$, operates on each of the points and normals in $\mathcal{D}$ according to the rules shown in (3).

$$\begin{aligned} \mathbf{Tp} &= \mathbf{Rp} + \mathbf{t} \\ \mathbf{Tn} &= \mathbf{Rn} \end{aligned} \tag{3}$$

3 Sherd Data Generation

We assume sherds are generated as follows. Nature generates a number of pots of various shapes, breaks each pot into fragments along break curves (Fig. 2) she has drawn on the surface, scatters a subset of each such set of fragments, and also scatters some pot-like fragments that do not come from pots. Our job is to estimate mathematical models of the original pots from laser scans of these sherds.

Sherd measurement-data is provided by a Shapegrabber laser/camera scanner [1]. It produces 15,000 3D points/sec. at a resolution and accuracy of the order of 0.25mm. All of these points are surface measurements, *i.e.*, measurements of outer, inner, and break surfaces including the 3D ridges that separate these surfaces. For the algorithms used in this

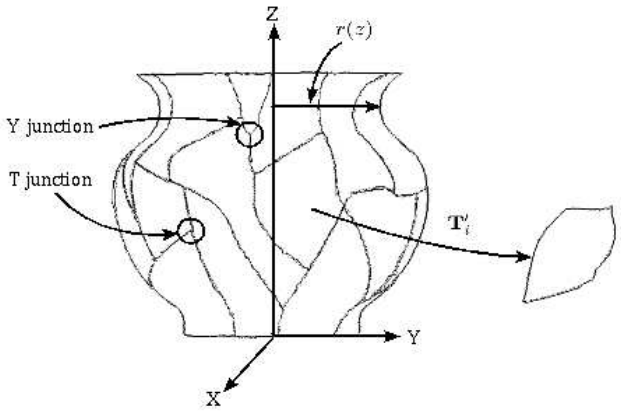

Figure 2: A fragmented vessel and it's parametric description.

paper, we have extracted two subsets of the measurement data: (1) those points which are measurements of a sherd outer-surface, denoted $\mathcal{D}_i$ for the i^{th}sherd, and (2) those which are measurements of a sherd outer-surface break-curve, denoted $\mathcal{P}_i$ for the i^{th} sherd. At present, using a computer, a person selects a set of possible locations of Y and T junctions from the sherd boundary data which we refer to as vertices. In the future, we will do this automatically.

Using the selected vertices the sherd break-curve, $\mathcal{P}_i$, is divided into parts. Each part, referred to as a break-point segment, describes a contiguous piece of the measured break-curve in the vicinity, i.e., within a radius R, of the hypothesized vertex. Data for each segment is extracted according to our simple parameterization of the unknown pot break-curves adopted in § 2. Hence, a break-point segment data is simply a special sequence of K measured 3D points and outer surface normals (see the discussion on break-curve parameters in § 2 and Fig. (3) for clarification). Each break-point data segment is assigned a unique index, the v^{th} break-point data segment is written $\mathcal{B}_v = ((\mathbf{p}_{v,1}, \mathbf{p}_{v,2}, \dots \mathbf{p}_{v,K}), (\mathbf{n}_{v,1}, \mathbf{n}_{v,2}, \dots, \mathbf{n}_{v,K}))$ where $\mathbf{p}_k$ denotes the k^{th} 3D break-point and $\mathbf{n}_k$ denotes the k^{th} 3D surface normal for $\mathcal{B}_v$.

3.1 Assumptions

Surface measurement points are *i.i.d.* $\mathbf{N}(0, \sigma_{\mathcal{D}}^2)$
These are independent, identically distributed, Gaussian perturbations perpendicular to the surface and having mean 0 *and variance* σ_s^2. See [2] for a justification of this model.
Surface measurement normals are *i.i.d.* $\mathbf{N}(0, \lambda\sigma_{\mathcal{D}}^2\mathbf{I})$
These are independent, identically distributed circularly symmetric Gaussian perturbations on the unit sphere , i.e., in SO(3), about each measured normal on the sherd surface, with mean 0 *and variance* $\lambda\sigma_{\mathcal{D}}^2$.
Break-curve measurement points are *i.i.d.* $\mathbf{N}(0, \sigma_{\mathcal{B}}^2\mathbf{I})$
These are independent, identically distributed spherically symmetric Gaussian perturbations in 3-space about each point on the true break-curve, with mean 0 *and variance* $\sigma_{\mathcal{B}}^2$. Note that, more appropriate but more complicated models can be used.

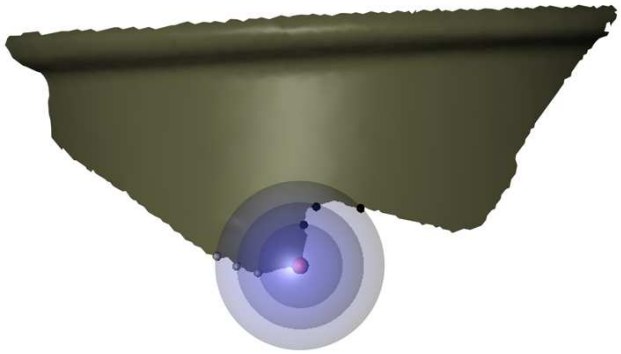

Figure 3: Break-point segments : a sherd outer surface, in grey, and one of the sherd vertices is shown as a large opaque red sphere. Two sets of break-point data which we call *break-point segments* are generated shown as light grey and black points on the break-curve. Each break-point segment has 4 ordered elements, starting with the vertex and then listed in order of increasing distance from the vertex. Note that break-point segment points lie at locations where a sphere (in transparent blue) of radius rR intersects the break-curve where $r = [1, 2, 3]$.

4 On Jointly Aligning Pairs of Sherds and Estimating Their Parameters

Sherd pairs are used as the basic building block of the assembly algorithm. To estimate parameters for a sherd pair, we hypothesize that two distinct sherds say sherds (i, j) share common break segments (m, n) respectively. Based on our model from § 3, we know that if this hypothesis is true, the break-point data segments are noisy measurements of a portion of the global break curve $\boldsymbol{\beta}$ and that the surface data from each sherd provide noisy estimates of the pot axis, $\mathbf{l}$, and a portion of the global profile curve $\boldsymbol{\alpha}$. To estimate these parameters, we arbitrarily assign the coordinate system of one of the two sherds, say sherd i, as the global coordinate system, hence $\mathbf{T}_i = (\mathbf{t} = (0, 0, 0), \mathbf{R} = \mathbf{I})$. We then estimate $\mathbf{T}_j$, the best alignment transformation between sherds i and j given that break-point segments m and n correspond, by computing the maximum likelihood esti-

from the Proceedings of CVPR (Conference on Computer Vision and Pattern Recognition) Vol I, June 2004

mates of $\mathbf{T}_j, \mathbf{l}, \boldsymbol{\alpha}, \boldsymbol{\beta}$ for the measurement data for sherds i and j. Toward this end, consider

$$\mathbf{P}(\mathcal{D}_i, \mathcal{D}_j, \mathcal{B}_{ci}, \mathcal{B}_{cj} | \mathbf{l}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{T}_j) \quad (4)$$

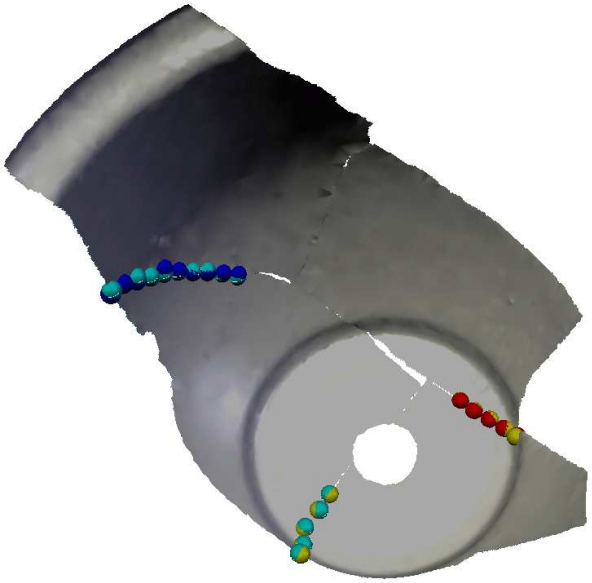

Figure 4: A correct configuration of 4 sherds : Matched break-point segments are indicated with spheres located on the sherd break curves. Note that we are using a subset of the available break-point segments to align the sherds. Different color spheres indicate break-point data from different sherds in the configuration.

We explain (4). Each of one or more break-point data segments in sherd i is matched with a break-point data segment in sherd j. This may involve only one break segment for each sherd, e.g., segments m and n for sherds i and j, respectively, or could involve all possible break segments that can occur for the two sherds with sherd j in position determined by $\mathbf{T}_j$ (see Fig. (4)). $\mathcal{B}_{ci}$ denotes the break-point segment data for sherd i that is being used in the sherds i and j matching and alignment. Similarly for sherd j. Conditioning the data probability on $\mathbf{l}, \boldsymbol{\alpha}, \boldsymbol{\beta}$ is on those portions of these parameters that are appropriate to the data sets $\mathcal{D}_i, \mathcal{D}_j, \mathcal{B}_{ci}, \mathcal{B}_{cj}$. Since the noises for the surface-point and break-point measurements are statistically independent, we have that (4) can be written as (5).

$$\mathbf{P}(\mathcal{D}_i | \mathbf{l}, \boldsymbol{\alpha}) \mathbf{P}(\mathcal{B}_{ci} | \boldsymbol{\beta}) \mathbf{P}(\mathcal{D}_j | \mathbf{l}, \boldsymbol{\alpha}, \mathbf{T}_j) \mathbf{P}(\mathcal{B}_{cj} | \boldsymbol{\beta}, \mathbf{T}_j) \quad (5)$$

$\mathbf{P}(\mathcal{D}_j | \mathbf{l}, \boldsymbol{\alpha}, \mathbf{T}_j)$ denotes the probability of the data set $\mathbf{T}(\mathcal{D}_j)$, i.e., $\mathcal{D}_j$ Euclidean transformed by $\mathbf{T}_j$, given the 3D surface parametrized by $\mathbf{l}$ and $\boldsymbol{\alpha}$. Similarly, $\mathbf{P}(\mathcal{B}_{cj} | \boldsymbol{\beta}, \mathbf{T}_j)$ is the probability of break-point data segments $\mathbf{T}_j(\mathcal{B}_{cj})$ which are measurements after their transformation by $\mathbf{T}_j$, given break-point parameters $\boldsymbol{\beta}$. Hence, we desire $\widehat{\mathbf{l}}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}, \widehat{\mathbf{T}}_j$ which are values of these parameters for which (5) is maximum. Unfortunately, this is a nonlinear problem and is computationally prohibitive to do for a large number of sherd pairs and many alignments for each pair. Hence, we solve a simpler problem (6) which corresponds to computing the MLE of a projection of the higher-dimensional joint distribution from (4).

$$\widetilde{\mathbf{T}}_j = \arg\max_{\mathbf{T}_j} \ln \left(\mathbf{P}\left(\mathcal{B}_{ci} | \boldsymbol{\beta}\right) \mathbf{P}\left(\mathcal{B}_{cj} | \boldsymbol{\beta}, \mathbf{T}_j\right) \right) \quad (6)$$

Given our noise assumptions for the break segment points and normals from § 3.1, the MLE solution to this problem is the optimization problem (7) where $\left(\sigma_{\mathcal{B}}^2, \lambda\sigma_{\mathcal{D}}^2\right)$ are constants related to the data noise pdfs from § 3.1.

$$\widetilde{\mathbf{T}}_j = \min_{\mathbf{T}_j} \frac{1}{2K} \sum_{k=1}^{K} \left[\frac{1}{\sigma_{\mathcal{B}}^2} \left\| \mathbf{p}_{l,k} - \mathbf{T}_j \mathbf{p}_{m,k} \right\|^2 + \frac{1}{\lambda\sigma_{\mathcal{D}}^2} \left\| \mathbf{n}_{l,k} - \mathbf{T}_j \mathbf{n}_{m,k} \right\|^2 \right] \quad (7)$$

Unlike maximizing (5), equation (7) is a linear least-squares problem and has an *explicit* solution which may be computed at little cost, for details see [6, 10, 13]. We then compute the most probable value of the parameters given the transformation $\widetilde{\mathbf{T}}_j$ by solving (8).

$$\widetilde{\mathbf{l}}, \widetilde{\boldsymbol{\alpha}}, \widetilde{\boldsymbol{\beta}} = \arg\max_{\mathbf{l}, \boldsymbol{\alpha}, \boldsymbol{\beta}} \ln \left(\mathbf{P}\left(\mathcal{D}_i, \mathcal{D}_j, \mathcal{B}_{ci}, \mathcal{B}_{cj} | \mathbf{l}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \widetilde{\mathbf{T}}_j \right) \right) \quad (8)$$

Why is this a reasonably useful result? In matching two break-point data segments, one for each sherd, we are not only computing the distances between pairs of break-point measurements, one of each sherd, but we are also comparing the measured surface-normals at these points. Hence, some surface measurement information is involved in the estimate $\widetilde{\mathbf{T}}_j$, and $\widetilde{\mathbf{T}}_j$ is a useful approximation to $\widehat{\mathbf{T}}_j$, and $\widetilde{\mathbf{l}}, \widetilde{\boldsymbol{\alpha}}, \widetilde{\boldsymbol{\beta}}$, are useful approximations to $\widehat{\mathbf{l}}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}$, respectively.

The pair is then assigned a preliminary cost called the *match cost* which is the negative log-likelihood of the aligned sherd data given the estimated pot parameters as shown in (9).

$$\epsilon_{(i,j),(m,n)} = -\ln \left(\mathbf{P}\left(\mathcal{D}_i, \mathcal{B}_{ci} | \widetilde{\mathbf{l}}, \widetilde{\boldsymbol{\alpha}}, \widetilde{\boldsymbol{\beta}} \right) \mathbf{P}\left(\mathcal{D}_j, \mathcal{B}_{cj} | \widetilde{\mathbf{l}}, \widetilde{\boldsymbol{\alpha}}, \widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{T}}_j \right) \right) \quad (9)$$

We assume that the match cost is a unique value associated with matching sherd (i, j) along break segments (m, n) (see § 5.1 for justification). In summary, although our quick solution is sub-optimal, we exploit the computational speed to filter out matches which are obviously wrong. Additionally, match costs are necessary for detecting configurations of sherds which contain the same set of matches, see § 5.1.

Eventually we do compute the MLE solution to (4) (see § 7), but only when the pair is chosen for assembly into a larger configuration.

4.1 A word on multiscale boundary matching

Since pairs of sherds may match along short or long break segments there is a need for multi-scale boundary curve matching. The parameter R, which denotes the distance from a sherd vertex to the K^{th} break point in a break-point segment defines the largest desirable scale, i.e., number of points in a break-point segment, for matching. Shorter break segments consist of sequential sets of elements from $\mathcal{B}_v$ starting at the first element. For a set of scales, $s = (1, 2, \ldots, S)$, the break segment defined for scale s has elements $\mathcal{B}_{v,s} = \left(\left(\mathbf{p}_1, \mathbf{p}_2, \ldots \mathbf{p}_{\frac{sK}{S}}\right), \left(\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_{\frac{sK}{S}}\right)\right)$, where v denotes the break segment index and s denotes the scale. *Our system computes match costs for all pairs of sherd break segments at all scales and stores the best set of break parameters for each pair*. Since our approximate method is very fast, we can do this exhaustively even for many pieces and at several different scales. For example, assume that there are 50 sherds and that on average each sherd has 4 vertices and we define 3 different matching scales. This results in approximately $\begin{pmatrix} 50 \\ 2 \end{pmatrix} * 8^2 * 3 = 235200$ comparisons which require approximately 4 minutes on contemporary CPUs which can solve (7) for $K = 5$ in less than 1ms. Note, a break-curve begins and ends on a vertex. For each break-curve there are two break-point segments, each beginning at one of the two vertices. Hence, there are 8^2 break-point segment matches to be computed at each scale for a pair of sherds.

5 Configurations of N Sherds (N > 2)

This section shows how MLE estimation of joint geometries of sherd pairs extends to configurations of size N. Let's consider a set of N sherds with indices $I = \{i_1, i_2, \ldots, i_N\}$. For simplicity, renumber these as $1, 2, \ldots N$. As in § 4, we begin by selecting the coordinate system of sherd 1 as the global coordinate system of the virtual pot *(actually, in practice we take the coordinate system of the sherd with smallest match cost)*, hence $\mathbf{T}_1 = (\mathbf{t} = (0,0,0), \mathbf{R} = \mathbf{I})$. We must now estimate the $N-1$ transformations $\mathbf{T}_{N-1} = \{\mathbf{T}_2, \mathbf{T}_3, \ldots, \mathbf{T}_N\}$ which position the remaining sherds in the virtual pot. This should be done by maximizing

$$\widehat{\mathbf{T}}_{N-1}, \widehat{\mathbf{l}}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}} = \arg\max_{\mathbf{T}_{N-1}, \mathbf{l}, \boldsymbol{\alpha}, \boldsymbol{\beta}} \ln\left(\mathbf{P}\left(\mathcal{D}_1, \mathcal{B}_{c1} | \mathbf{l}, \boldsymbol{\alpha}, \boldsymbol{\beta}\right) \prod_{w=2}^{N} \mathbf{P}\left(\mathcal{D}_w, \mathcal{B}_{cw} | \mathbf{l}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{T}_w\right)\right) \tag{10}$$

Optimizing (10) simultaneously with respect to all $6(N-1)$ parameters of $\mathbf{T}_{N-1}$ and $\mathbf{l}, \boldsymbol{\alpha}, \boldsymbol{\beta}$ is computationally expensive and prone to failure, i.e., hitting a local minima, due to dependencies between the matched parameters for each sherd. Hence, we proceed using our approximate pairwise solution $\widetilde{\mathbf{T}}_2, \widetilde{\mathbf{l}}, \widetilde{\boldsymbol{\alpha}}, \widetilde{\boldsymbol{\beta}}$ from § 4 as a initial values to compute the MLE estimates $\widehat{\mathbf{T}}_2, \widehat{\mathbf{l}}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}$ in (11). This involves a number of iterations in a nonlinear maximization. Now estimate $\mathbf{T}_3$ and update $\widehat{\mathbf{l}}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}$, but not $\widehat{\mathbf{T}}_2$— it remains fixed. This is done by

$$\begin{aligned} \check{\mathbf{T}}_3, \check{\boldsymbol{\beta}} \quad = \arg\max_{\mathbf{T}_3, \boldsymbol{\beta}} \ln\Big(&\mathbf{P}(\mathcal{D}_1, \mathcal{B}_{c1} | \widehat{\mathbf{l}}, \widehat{\boldsymbol{\alpha}}, \boldsymbol{\beta}) \times \\ &\mathbf{P}(\mathcal{D}_2, \mathcal{B}_{c2} | \widehat{\mathbf{l}}, \widehat{\boldsymbol{\alpha}}, \widehat{\mathbf{T}}_2, \boldsymbol{\beta}) \times \\ &\mathbf{P}(\mathcal{D}_3, \mathcal{B}_{c3} | \widehat{\mathbf{l}}, \widehat{\boldsymbol{\alpha}}, \mathbf{T}_3, \boldsymbol{\beta})\Big) \end{aligned} \tag{11}$$

Note, the portion of $\boldsymbol{\beta}$ that changes is only the break-point segments associated with the active break-point data segments for sherd 3. Included in (11) is a term involving how well $\mathcal{D}_3$ fits the hypothesized pot surface specified by $\widehat{\mathbf{l}}, \widehat{\boldsymbol{\alpha}}$. Now update $\widehat{\mathbf{l}}, \widehat{\boldsymbol{\alpha}}$ by keeping $\mathbf{T}_2$ and $\mathbf{T}_3$ fixed at $\widehat{\mathbf{T}}_2$ and $\check{\mathbf{T}}_3$, respectively, and estimating the best axis profile curve, $\check{\mathbf{l}}, \check{\boldsymbol{\alpha}}$ for the three fixed data sets $\mathcal{D}_1, \widehat{\mathbf{T}}_2(\mathcal{D}_2), \check{\mathbf{T}}_3(\mathcal{D}_3)$.

$$\begin{aligned} \check{\mathbf{l}}, \check{\boldsymbol{\alpha}} \quad = \arg\max_{\mathbf{l}, \boldsymbol{\alpha}} \ln\big(&\mathbf{P}(\mathcal{D}_1, \mathcal{B}_{c1} | \mathbf{l}, \boldsymbol{\alpha}, \check{\boldsymbol{\beta}}) \times \\ &\mathbf{P}(\mathcal{D}_2, \mathcal{B}_{c2} | \mathbf{l}, \boldsymbol{\alpha}, \check{\boldsymbol{\beta}}, \widehat{\mathbf{T}}_2) \times \\ &\mathbf{P}(\mathcal{D}_3, \mathcal{B}_{c3} | \mathbf{l}, \boldsymbol{\alpha}, \check{\boldsymbol{\beta}}, \check{\mathbf{T}}_3)\big) \end{aligned} \tag{12}$$

Now repeat for the inclusion of sherd 4 to obtain estimates $\check{\mathbf{T}}_4, \check{\mathbf{l}}, \check{\boldsymbol{\alpha}}, \check{\boldsymbol{\beta}}$. Note that $\check{\mathbf{l}}, \check{\boldsymbol{\alpha}}, \check{\boldsymbol{\beta}}$ are not quite the MLE's $\widehat{\mathbf{l}}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}$ based on data for N sherds, but they are close, and MLE's can be obtained for an additional computational cost.

Each assembled sherd group is assigned a *configuration cost* (13) which represents the probability of the configuration given the connected, i.e. hypothesized subset, of break segment correspondences and the sherd surface data. The cost is used to determine the order of comparisons for the search algorithm (see § 6).

$$\epsilon_{\cup_w(w,cw)} = -\ln\left(\mathbf{P}\left(\mathcal{D}_1, \mathcal{B}_{c1} | \widehat{\mathbf{l}}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}\right) \prod_{w=2}^{N} \mathbf{P}\left(\mathcal{D}_w, \mathcal{B}_{cw} | \widehat{\mathbf{l}}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}, \widehat{\mathbf{T}}_w\right)\right) \tag{13}$$

Note that the hypothesized set of sherd and break segment correspondences provides a unique index, $\cup_{w=1}^{N}(w, cw)$, and hence, the order of the hypotheses is not considered important. This means that the configuration of pieces built from a set of hypotheses, e.g., the triplet $h_1 = \{A, B, C\}$, is indistinguishable from any permutation of that set, e.g., $h_2 = \{B, A, C\}$.

5.1 Detecting Duplicate Configurations

Permuted correspondence hypotheses are simply different sequences of constructing the same configuration by matching the same break segments in a different order. Instances of hypothesis sets are detected trivially using the *match costs* computed in § 4. Since each match cost represents the value of a continuous random variable, i.e., the exponent of the pdf from (8), the probability that two match costs are the same is simply 0, which allows us to index a specific hypothesis with a unique match cost. Hence, sets of hypotheses are simply represented as sums of match costs which also have this uniqueness property. All permutations of a set of hypotheses add to the same constant. Hence, when we consider merging a sherd with a configuration we compute the match cost of the new set of hypotheses and simply check if the match cost is a member of the set of previously computed match costs, i.e., "Have I already constructed a permutation of this hypothesis?". If so, then we discard the hypothesized match, else, we construct the new configuration.

6 Pot Assembly Search Algorithm

The sherds may only describe a single or several small portions of the overall vessel. Therefore, the pot assembly algorithm must accomodate for situations where some sherds may be missing or sherds come from a number of different vessels.

To this end, we propose a MLE-based algorithmic search in order to robustly perform pot assembly. The algorithm examines *significant* configurations of sherds according to the method described in § 5. Here and afterwards *significant* denotes those configurations whose joint-data probability, equivalently, *configuration cost* (13), represents a possible (i.e. not-improbable) solution. We proceed as follows:

1. Estimate the axis/profile-curve for each of the sherds § 2 (*Computationally fast*).

2. For each pair of sherds, compute all reasonable alignments of their break segments § 4 (at all scales § 4.1) (*Computationally fast*).

3. For each significant configuration, improve the alignment using § 5 and store configurations and individual sherds in order of increasing cost in a stack (*Computationally of medium cost at present*).

4. Starting with the top item in the stack, go down through the stack and merge the configuration with those lower in the stack that result in roughly lowest cost configurations according to § 5. Update the stack. Note, a sherd can appear only once in a configuration, though the same sherd can appear in many configurations. Return to step 3 or stop (*Computationally of medium cost at present*).

Note, this search proceeds along contours of constant configuration probability to find the most probable virtual pot for all sizes up to N. Hence realizations of high probability will perpetuate through the search algorithm whereas improbable geometries will eventually not be considered. In practice, we limit the number of configurations considered for each size which makes the search problem proceed faster. For our results, the number of allowed configurations, Q, of size N is $Q = 60, 10, 20, 20, 20, 20, 50, 80, 100$ for $N = 2, 3, 4, \ldots, 10$, respectively. This allows for correct assembly of a 10 sherd pot in 1.75 hours and the search is completed in less than 5 hours.

7 Sherd Alignment

This section is details of our solution to sherd pair alignment, i.e., maximization of equation (4) and alignment of configurations involving N sherds, i.e., equation (10). Since the covariance matrix for the sherd data are the same for all sherds, maximizing (4,10) corresponds to minimizing *only* the exponential term of the sherd joint distribution. For aligning the sherd i_k surface data and its connected break-curve segment data $\mathcal{X}_{i_k,c} = (\mathcal{D}_{i_k}, \mathcal{B}_c)$ to the surface and break-curve segment data for sherd (or sherd configuration) i_1, this simplifies to the equation shown in (14).

$$e_{i_k} = \arg\min_{\mathbf{T}_{i_k}} \frac{1}{6CK} \|\mathbf{T}_{i_k}(\mathcal{B}_c) - \mathcal{B}_{i_1,c}\|^2 + \frac{1}{2\sigma_\alpha^2} \alpha(\mathbf{T}_{i_k}(\mathcal{D}_{i_k}) \cup \mathcal{D}_{i_1})^2 \tag{14}$$

In words, we seek to find the 3D transformation which minimizes the sum of 2 terms : (1) the squared distance between the transformed points and normals on the matched break-curve segments of the sherd pair (i_k, i_1), and (2) *the average approximate squared Euclidean distance, i.e., squared algebraic distance, of the transformed sherd i_k surface data and the sherd i_1 surface data to the axially symmetric virtual pot surface defined by the axis/profile-curve pair* $(\mathbf{l}, \boldsymbol{\alpha})$. Solving equation (14) is an alignment problem where the term in β corresponds to errors in the alignment of one or more break segments and the term in $\alpha(\mathcal{D})$ corresponds (approximately) to the 3D distance of the sherd surface dataset to the virtual pot surface, i.e., the axially symmetric 3D surface defined by the axis/profile curve pair $(\mathbf{l}, \boldsymbol{\alpha})$.

Hence, we can compute the gradient of the algebraic distance (note we apply the chain rule) and efficiently solve for the transformation using a Levenberg-Marquardt minimization scheme. This could be considered a variant of the "fast-ICP" method proposed in [3] where we replace the distance transform with the algebraic distance and we also add an

error term involving the break parameters. We have found this alignment scheme to be robust to noise and relatively efficient to compute, e.g., for 3000 data points convergence is achieved within 20 iterations which requires about 5 seconds of CPU time in total.

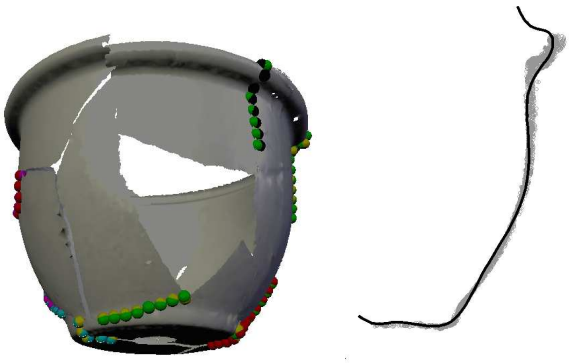

Figure 5: A correctly assembled pot of 13 sherds where only the 10 matched sherds shown were available.

8 Conclusion

A Bayesian approach has been outlined for the estimation of mathematical representations for pots based on sherds found at archaeology sites. The framework discussed in this paper is for estimating arbitrary *a priori* unknown axially-symmetric pot models. Hence, it is *unsupervised* pot geometry-learning from sherd data. If instead we know *a priori* that the pot sherds present are not arbitrary but rather that each belongs to one of a group of 10 known pot shapes e.g., the problem is computationally much easier because the sherd alignment problem is then more of a pot shape-recognition problem and less of a shape-estimation problem. The framework presented can accommodate additional geometric and pattern information which should result in doing the pot estimation faster, or with fewer sherds, or estimating models for more complex objects.

The problem is conceptually and computationally difficult because we are estimating continuous geometry from noisy estimates of fragments of continuous geometries. Using all the available information at each stage is not an option since that would be computationally prohibitive. Rather, of importance is to use appropriate partial information at each stage. We view our approach as having reached an important conceptual and computational stage, but additional work is necessary to : better use all possible information available; assemble pots broken into very large numbers of sherds, e.g., 40; make use of very small sherds; treat completely free-form shapes such as sculpture, column capitals, etc. This material is based upon work supported by the National Science Foundation Grant No. 0205477.

References

[1] . Shapegrabber inc. http://www.shapegrabber.com.

[2] R. Bolle and D. B. Cooper. On optimally combining pieces of information, with application to estimating 3-D complex-object position from range data. *IEEE Trans. on Pattern Anal. Machine Intell.*, 8(5):619–638, 1986.

[3] A. Fitzgibbon. Robust Registration of 2D and 3D Point Sets. In *Proc. of the British Machine Vision Conference*, 2001.

[4] H. Freeman and L. Garder. A Pictorial Jigsaw Puzzles: The Computer Solution of a Problem in Pattern Recognition. *IEEE Trans. Electron. Comput.*, 13:118–127, 1964.

[5] D. Goldberg, C. Malon, and M. Bern. A Global Approach to Automatic Solution of Jigsaw Puzzles. In *Proc. of Conf. on Computational Geometry*, pages 82–87, 2002.

[6] B. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal Optical Society of America*, 4(4):629–642, 1987.

[7] W. Kong and B. B. Kimia. On solving 2D and 3D puzzles using curve matching. In *Proc. of CVPR*, Hawaii, USA, December 2001. IEEE, Computer Society.

[8] S. Li. *Markov Random Field Modeling in Image Analysis*, chapter 7. Springer, 2001.

[9] G. Papaioannou, E.-A. Karabassi, and T. Theoharis. Reconstruction of three-dimensional objects through matching of their parts. *PAMI*, 2002.

[10] X. Pennec and J. Thirion. A framework for uncertainty and validation of 3-d registration methods based on points and frames. *IJCV*, 1997.

[11] J. Stolfi and H. Leitão. A multiscale method for the reassembly of two-dimensional fragmented objects. *PAMI*, 2002.

[12] G. Ucoluk and I. H. Toroslu. Automatic Reconstruction of Broken 3-D Surface Objects. *Comp. and Graph.*, 23(4):573–582, 1999.

[13] S. Umeyama. Least-Squares Estimation of Transformation Parameters Between Two Point Patterns. *PAMI*, 13(4):376–380, 1991.

[14] A. Willis. *Stochastic 3D Geometric Models for Classification, Deformation, and Estimation*. PhD thesis, Brown University, to Appear - 2004.

[15] A. Willis, D. Cooper, et al. Assembling virtual pots from 3D measurements of their fragments. In *VAST International Symposium on Virtual Reality Archaeology and Cultural Heritage*, pages pp. 241–253, 2001.

[16] A. Willis, D. Cooper, et al. Bayesian Pot-Assembly from Fragments as Problems in Perceptual-Grouping and Geometric-Learning. In *ICPR*, volume III, pages 297–302, 2002.

[17] A. Willis, D. Cooper, et al. Accurately Estimating Sherd 3D Surface Geometry with Application to Pot Reconstruction. In *CVPR Workshop : ACVA*, June 2003.

[18] H. Wolfson, E. Schonberg, A. Kalvin, and Y. Lambdan. Solving Jigsaw Puzzles Using Computer Vision. *Annals of Operations Research*, 12:51–64, 1988.