AUTOMATIC REASSEMBLY OF THREE–DIMENSIONAL JIGSAW PUZZLES

ANNA GRIM, TIMOTHY O'CONNOR, PETER J. OLVER, CHEHRZAD SHAKIBAN, RYAN SLECHTA, AND ROBERT THOMPSON

Abstract. In this paper, we present an effective algorithm for reassembling three-dimensional apictorial jigsaw puzzles obtained by dividing a curved surface into a finite number of interlocking pieces. As such, our algorithm does not make use of any picture or design that may be painted on the surface; nor does it require a priori knowledge of the overall shape of the original surface. A motivating example is the problem of virtually reconstructing a broken ostrich egg shell.

In order to develop and test the algorithm, we also devise a method for constructing synthetic three-dimensional puzzles by randomly distributing points on a compact surface with respect to surface area measure, then determining the induced Voronoi tessellation, and finally curving the Voronoi edges by using Bezier curves with selected control points.

Our edge-matching algorithm relies on the method of Euclidean signature curves. The edges of the puzzle pieces are divided into bivertex arcs, whose signatures are directly compared. The algorithm has been programmed in Matlab and is able to successfully reassemble a broad range of artificial puzzles, including those subjected to a reasonable amount of noise. Moreover, significant progress has been made on reassembly of the real-world ostrich egg data.

Keywords: jigsaw puzzle, curvature, torsion, Euclidean signature, bivertex arc, Voronoi tessellation, Bezier curve.

"Yes, all his horses and all his men," Humpty Dumpty went on.
"They'd pick me up again in a minute, they *would!"*

— Lewis Carroll, *Through the Looking–Glass and What Alice Found There*

Date: April 7, 2016.

1. Introduction

In this paper, we develop a novel algorithm for the assembly of three-dimensional jigsaw puzzles obtained by dividing a compact curved surface $S \subset \mathbb{R}^3$ into a finite number of non-overlapping, connected, relatively open subsets, called *pieces*. This may be viewed as the generalized "Humpty Dumpty" problem, the original being the reassembly of a broken egg shell. Our algorithm handles *apictorial puzzles*, meaning that it only involves the shape of the puzzle pieces, and does not make use of any picture or design which may be painted on its surface. It does not require any a priori knowledge of the overall shape of the original surface since it directly matches the edges of the pieces considered as space curves and not as curves embedded in a known surface, and hence can, in principle, reconstruct puzzles obtained from arbitrary curved surfaces, including those with nontrivial topology, e.g. tori. Alternative approaches to this problem can be found, for example, in [20, 21, 38]. We do not investigate the more challenging problem of reassembling three-dimensional solid objects, including broken statues and other archaeological artifacts, [33], and also bone fractures, [37].

The assembly of flat two-dimensional jigsaw puzzles has been studied by a number of authors. Most solution algorithms, [10, 40, 41], have been restricted to archetypical puzzles, in which the pieces are of a rectangular form, with indents and outdents on edges interior to the puzzle, positioned on an (approximate) rectangular grid, and, typically, with a rectangular outer boundary. In contrast, the signature-based algorithm developed in [15] works with arbitrarily shaped pieces and irregular puzzle boundaries — indeed the assembly usually starts in the middle of the puzzle. This algorithm was able to successfully reassemble some challenging commercially available planar jigsaw puzzles, including the 69 piece Baffler Nonagon, [42], and thus holds much promise for applications well beyond realm of recreational jigsaw puzzles. The fact that the puzzle is apictorial implies that the similarity measure used to match the pieces relies solely on the geometric shapes of their edges. The algorithm is based on matching the Euclidean signature curves, [3], of the bivertex arcs, [14], contained in the puzzle piece edges, which can be viewed as the curve's "signature codons". It relies on the fact that the Euclidean signature is invariant under rotations and translations of the original curve and, moreover, uniquely characterizes suitably non-degenerate curves up to such rigid motions.

The edges of puzzle pieces taken from a curved surface $S \subset \mathbb{R}^3$ are space curves. The problem of curve matching in three-dimensional space was perhaps first investigated by Kishon and Wolfson, [19], who employed the classical result that a space curve is uniquely determined, up to rigid motions, by its curvature and torsion invariants as functions of arc length, cf. [12]. However, there are several practical difficulties that complicate using this characterization of rigidly equivalent curves. To start with, the arc length is ambiguously defined, since it depends on the choice of an initial point on the curve. Thus, one must identify two curvature functions that differ by a translation, $\kappa(s+c) \simeq \kappa(s)$, and similarly for torsion. This becomes increasingly problematic if one needs to deal with occlusions or, as in the case of jigsaw puzzle pieces, only match parts of the space curves.

Instead, we follow [3, 30], and use the Euclidean differential invariant signature curves to match the puzzle piece boundaries. The *signature* of a space curve $C \subset \mathbb{R}^3$ is the space curve $\Sigma \subset \mathbb{R}^3$ parametrized by the curvature κ, the derivative of curvature with respect to arc length, κ_s, and the torsion τ. The key fact is that two suitably regular space curves C and $\widetilde{C}$ are equivalent under a Euclidean transformation, meaning that $\widetilde{C} = g \cdot C$ for some rigid motion g of $\mathbb{R}^3$, if and only if they have identical signature curves: $\widetilde{\Sigma} = \Sigma$. Observe

that the signature curve is entirely local and can be written in terms of any convenient parametrization of the curve by using well-known formulas for these fundamental Euclidean differential invariants, [12]. Furthermore, it is worth noting that the method of differential invariant signatures extends to surfaces and even higher dimensional submanifolds under general transformation groups, [29, 30], where there is no longer any fully invariant analogue of a distinguished arc length parametrization. Thus, the differential invariant signature approach has the theoretical capability of solving the more challenging problems of three-dimensional solid object reconstruction, although various significant practical details remain to be resolved.

This line of research was inspired by the scanned data of a broken ostrich egg shell that was sent to the third author by Marshall Bern, [1], who then posed the challenge of automatically reconstructing the egg. As we discuss in the final section, some encouraging progress has been made in this direction. However, in this initial report, we will not address the general problem of reconstructing broken surface objects obtained from scanned data. Indeed, at present, in contrast to the planar case, [15], obtaining useful real-world three-dimensional datasets remains a challenge, and is the subject of ongoing investigation.

Consequently, our automatic solution algorithm will be primarily tested on artificially generated puzzles. We thus begin by developing a methodology for generating random three-dimensional jigsaw puzzles, which will be based on geodesic Voronoi diagrams obtained by distributing points in a fairly uniform manner on the surfaces. For simplicity, we concentrate on generating puzzles on spherical and ellipsoidal surfaces, although, as noted above, our reconstruction techniques are completely independent of the overall shape of the surface.

The MATLAB code used to generate and reassemble synthetic jigsaw puzzles on spheres and ellipsoids can be found on the third author's web site:

`http://www.math.umn.edu/~olver/matlab.html`

2. Puzzle Generation

As noted in the introduction, the first stage of the project is to test our signature-based algorithm on synthetically generated three-dimensional jigsaw puzzles that are constructed by dividing a curved surface $S \subset \mathbb{R}^3$ into a finite number of non-overlapping pieces.

The surface S is assumed to be compact (closed and bounded, perhaps with boundary). Each piece of the puzzle will be based at a randomly chosen point on S, roughly representing the piece's centroid. In order to keep the relative sizes of the individual pieces comparable, we begin by selecting a prescribed number of points that are more or less uniformly distributed with respect to surface area. It is worth noting that powerful algorithms for distributing points on general curved surfaces based on graph Laplacians have been recently developed in [24], although we do not need such sophisticated techniques for our relatively simple problem.

In fact, we will be content to mostly concentrate on the simplest case in which the surface $S \subset \mathbb{R}^3$ is the unit sphere centered at the origin. The problem of uniformly distributing points on a sphere has been investigated in depth by many authors, with applications ranging from numerical quadrature, to molecular structure, to the positioning of satellites in geocentric orbits, [23, 32]. Equidistribution of points can be based on minimizing an appropriate total interpoint energy. For example, the *Fekete problem* is to find the point configuration that minimizes the electrostatic energy between uniformly charged point masses lying on the sphere, and is one of Smale's celebrated 18 problems for the twenty-first century, [35].

To generate the desired points, we will use the spiral point algorithm introduced by Rakhmanov, Saff, and Zhou, [31], which was motivated by hexagonal tiling techniques. We divide the sphere into n equally spaced latitudinal circles, including the north and south poles. Each circle will contain precisely one spiral point. Starting with the south pole, the k-th spiral point $q_k = (\theta_k, \varphi_k)$, written in spherical coordinates — where we use the mathematical convention that θ_k is the *azimuthal angle* (longitude) and φ_k the *zenith angle* (latitude) — is obtained from the $(k-1)$-st point by first going upwards along a great circle (meridian) to the k-th circle and then traveling in a clockwise direction along the k-th circle for a distance

$$d = \frac{3.6}{\sqrt{n}}\,.$$

This value was motivated by Habich and van der Waerden's results, [13], on best packing, although their numerical constant $(8\,\pi)^{1/2}/3^{1/4} \approx 3.809$ has been slightly decreased to 3.6 following numerical experimentation, [32]. In summary, the spiral points $q_k = (\theta_k, \varphi_k)$ are recursively defined by

$$\begin{aligned} &h_j = -1 + \frac{2(j-1)}{n-1}, && \varphi_j = \arccos(h_j), && 1 \le j \le n, \\ &\theta_1 = \theta_n = 0, && \theta_k = \theta_{k-1} + \frac{d}{\sqrt{1-h_k^2}} \mod 2\pi, && 2 \le k \le n-1. \end{aligned} \tag{1}$$

The result of this construction is a set of $n \ge 2$ sample points $q_1, \dots, q_n \in S$ that are approximately uniformly distributed on the surface of the unit sphere.

To complete our spherical jigsaw puzzle, we need to construct non-overlapping pieces surrounding each sample point. The first step is to construct the resulting spherical Voronoi tessellation so that each q_k is the center point of a Voronoi cell. Suppose q_i and q_j belong to adjacent Voronoi cells. Their common Voronoi edge consists of points that lie equidistant from the two center points, under the geodesic (great circle) distance:

$$0 < d(q_i, q_j) = \arccos(q_i \cdot q_j) \le \pi.$$

The corresponding *Voronoi vertices* lie at the intersection of the boundaries of three or more Voronoi cells, [6, 28].

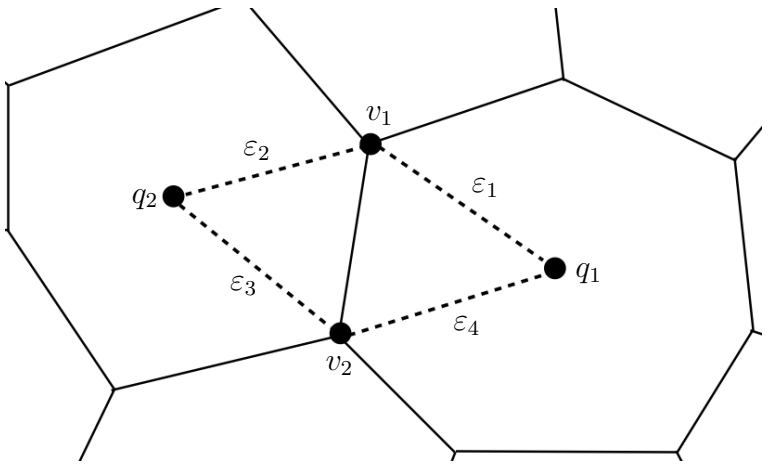

FIGURE 1. Voronoi Quadrilateral

The second step is to replace the circular Voronoi edges by suitably curved edges mimicking those found on a broken object or three-dimensional jigsaw puzzle. It must be ensured that the resulting curved edges do not intersect. To this end, consider the spherical quadrilaterals whose vertices are the center points of two adjacent Voronoi cells, say q_1, q_2, along with the Voronoi vertices, v_1, v_2, of their common Voronoi edge; see Figure 1. The edges of the spherical quadrilateral are denoted by ε_1, which joins q_1 to v_1, and ε_2, which joins v_1 to q_2, and ε_3, which joins q_2 to v_2, and ε_4, which joins v_2 to q_1. The puzzle piece edge will be obtained by replacing the Voronoi edge between v_1 and v_2 by the spherical projection

$$\beta(t) = \frac{\widetilde{\beta}(t)}{\| \widetilde{\beta}(t) \|} \tag{2}$$

of the Bézier curve parametrized by

$$\widetilde{\beta}(t) = (1-t)^3 p_0 + 3(1-t)^2 t\, p_1 + 3(1-t) t^2 p_2 + t^3 p_3, \qquad 0 \le t \le 1, \tag{3}$$

as prescribed by the four *control points* p_0, p_1, p_2, p_3, which are sequentially selected from the edges $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$ of the spherical quadrilateral. Since the Bézier curve (3) lies in the convex hull of its control points, [9], its spherical projection (2) will lie completely inside the spherical quadrilateral, and in this manner we are able to avoid any inadvertent intersection of the curved edges.

We are able to implement different edge styles by suitably modifying the Bézier control points. In all cases, we set $p_0 = v_1$, $p_3 = v_2$, so that the Bézier curve (3) connects the two edge vertices. To create an indented or outdented edge like on a traditional jigsaw puzzle, the other two control points p_1, p_2 are chosen randomly, with respect to arc length measure, from either two edges lying in the same Voronoi cell, e.g. ε_1 and ε_4, or from two edges on opposite sides of the quadrilateral, e.g. ε_2 and ε_4. See Figure 2 for examples of the resulting two principal types of curved edges employed here.

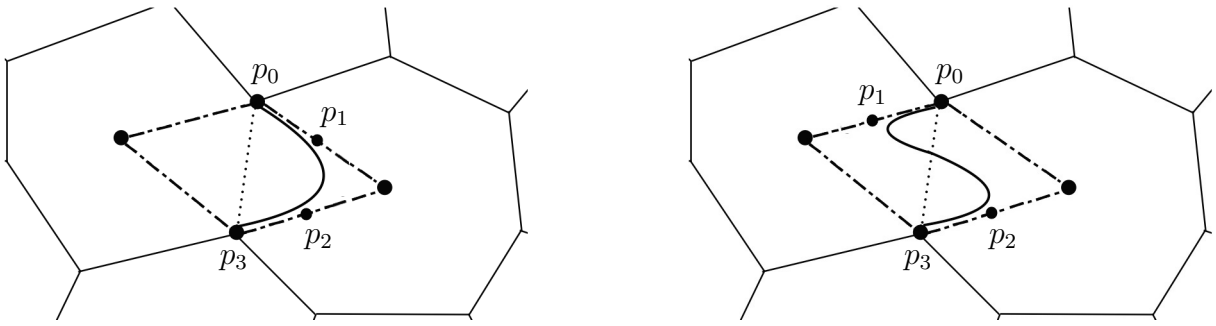

FIGURE 2. Bézier Control Points for Both Types of Curved Edges

Once the surface has been divided into uniformly sized pieces, the *puzzle data* is obtained by applying randomly generated rigid motions to the individual pieces. Thus, our goal is to reassemble the given puzzle data by applying suitable rigid motions to the individual pieces in order to match their edges and thereby reform the original surface. In Figure 3, we display one example of a synthetically generated spherical puzzle containing 9 pieces. The first plot shows the puzzle that our MATLAB code generated, and the second shows the individual pieces after being subjected to individual random rigid motions.

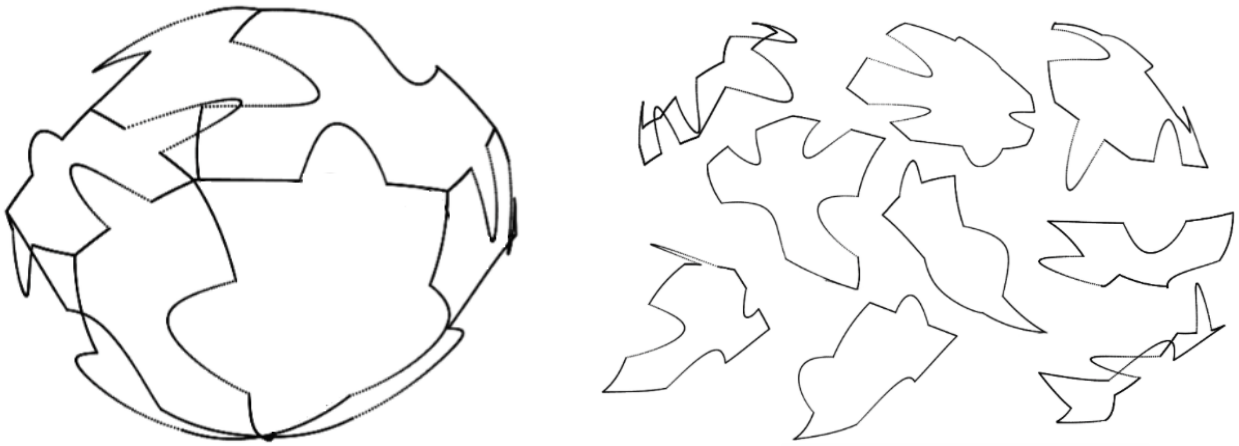

FIGURE 3. A Synthetic Puzzle

Our method of generating spherical jigsaw puzzles can be adapted to create puzzles on other genus zero closed surfaces by mapping them to a sphere. For example, an ellipsoidal jigsaw puzzle can be created by performing the scaling transformation,

$$T(x, y, z) = \begin{vmatrix} \lambda & 0 & 0 \\ 0 & \mu & 0 \\ 0 & 0 & \nu \end{vmatrix},$$

where $\lambda, \mu, \nu > 0$ are the semi-axes of the resulting ellipsoid. Since T is not area-preserving, the transformed points may no longer be as uniformly distributed on the ellipsoid with respect to surface area, and hence, depending upon the eccentricity, the resulting ellipsoidal puzzles may have less uniformly sized pieces.

For more general convex surfaces, one could employ the Gauss map, [36], that maps each point on the surface to the point on the unit sphere defined by its outward normal. Alternatively, one might employ the graph Laplacian algorithms developed in [24] to directly distribute the points on the surface, although this would then require developing a method of generating Voronoi tessellations on arbitrary curved surfaces. See also [7, 8, 27] for basic results on the method of graph Laplacians and their applications in image processing, data analysis, networks, and elsewhere.

3. Edge Matching

3.1. **3D Signature Curves.** The method of differential invariant signatures for comparing submanifolds up to group transformations was introduced to image processing in [3]. It is based on Élie Cartan's solution to the equivalence problem, [4], which demonstrates that two suitably regular submanifolds are locally equivalent under a prescribed Lie group action, meaning that they can be locally mapped to each other by a group transformation, if and only if they have identical functional interrelationships, known as *syzygies*, among all their differential invariants. In [29], Cartan's solution was reformulated in terms of differential invariant signatures, or, as they are called in that reference, classifying manifolds. The key fact is that the syzygies among the higher order differential invariants are all uniquely prescribed by a finite number of syzygies among certain low order fundamental differential

invariants, and the latter are used to parametrize the signature. This establishes the key result that two suitably regular submanifolds are locally equivalent if and only if they have identical differential invariant signatures. The equivariant method of moving frames, [30], can be employed to algorithmically determine a suitable set of signature invariants for very general Lie group actions on submanifolds.

In the case of plane curves $C \subset \mathbb{R}^2$ under the special Euclidean group SE(2), acting by rigid motions, i.e. translations and rotations, a complete list of differential invariants is provided by the curvature κ and its successive derivatives with respect to arc length: $\kappa_s, \kappa_{ss}, \dots$. Geometrically, κ measures the curve's deviation from linearity, while its derivative measures the rate of spiraling. The corresponding *Euclidean signature curve* Σ is parametrized by the first two of these differential invariants, so $\Sigma = \{(\kappa(t), \kappa_s(t))\}$, where t is any convenient parametrization of C. Assuming $\kappa_s \neq 0$, the signature curve Σ locally determines the syzygy

$$\kappa_s = F(\kappa)$$

relating the two fundamental differential invariants. Successive differentiation of this syzygy produces the corresponding syzygies among all the higher order differential invariants; for example, by the chain rule,

$$\kappa_{ss} = \frac{d}{ds}\kappa_s = \frac{d}{ds}F(\kappa) = F'(\kappa)\,\kappa_s = F'(\kappa)\,F(\kappa),$$

and similarly for the higher order curvature derivatives. This serves to justify our choice of κ, κ_s as the fundamental differential invariants parametrizing the signature curve.

In our case, the puzzle piece boundary curves lie in three-dimensional space. Under the special Euclidean group SE(3) of rigid motions, the basic differential invariants of a space curve $C \subset \mathbb{R}^3$ are its curvature κ and torsion τ, the latter measuring the curve's deviation from planarity, along with their successive derivatives with respect to the arc length element: $\kappa_s, \tau_s, \kappa_{ss}, \tau_{ss}, \dots$. It turns out that to parametrize a Euclidean signature, one only needs 3 of these invariants, namely $\Sigma = \{(\kappa(t), \kappa_s(t), \tau(t))\}$, where, as before, t is any convenient parametrization of C and one employs standard, well-known formulae for the curvature and torsion invariants, [12, 29]. Observe that we do not need to include τ_s since, assuming $\kappa_s \neq 0$, we can locally express $\kappa_s = F(\kappa)$, and $\tau = H(\kappa)$, which, by the chain rule, uniquely determines the syzygy

$$\tau_s = \frac{d}{ds}H(\kappa) = H'(\kappa)\,\kappa_s = H'(\kappa)\,F(\kappa).$$

As in the case of plane curves, the higher order syzygies can all be obtained by repeated differentiation, justifying the specification of fundamental signature invariants.

3.2. **Numerical Approximations.** Although the edges of our synthetic spherical jigsaw puzzles are smoothly parametrized Bézier curves, since our ultimate goal is to assemble puzzles obtained from scanned digital images, we will work directly with discretizations of these edges. Thus, approximate methods of calculating the signature invariants κ, κ_s, and τ are required in order to calculate the corresponding signature curve.

Let $p_1, \dots, p_n \in C$ be a discretization obtained by sampling a space curve $C \subset \mathbb{R}^3$ relatively uniformly with respect to arc length. According to [3], the curvature at the sample point p_i is invariantly approximated by the reciprocal of the radius of the circle passing through it and its two immediate neighbors, p_{i-1}, p_{i+1}, which, according to Heron's formula,

is

$$\widetilde{\kappa}(p_i) = 4\,\frac{\Delta}{abc} = 4\,\frac{\sqrt{s(s-a)(s-b)(s-c)}}{abc}\,. \tag{4}$$

Here

$$a = \| p_i - p_{i-1} \|, \qquad b = \| p_{i+1} - p_i \|, \qquad c = \| p_{i+1} - p_{i-1} \|, \tag{5}$$

are the side lengths of the triangle formed by the three points, while

$$s = \tfrac{1}{2}(a+b+c), \qquad \Delta = \sqrt{s(s-a)(s-b)(s-c)}\,, \tag{6}$$

are, respectively, its semiperimeter and area.

The derivative of curvature κ_s at p_i is calculated in a similar fashion by using the additional points $p_{i-2}, p_{i+2} \in C$. Assuming uniform spacing of the sample points p_i, the numerical approximation

$$\widetilde{\kappa}_s(p_i) = \frac{\widetilde{\kappa}(p_{i+1}) - \widetilde{\kappa}(p_{i-1})}{c}\,, \tag{7}$$

where c is defined in (5), was proposed and used in [3]. However, as noted by Boutin, [2], this formula produces inaccuracies when dealing with non-uniformly spaced sample points. Boutin found three alternative, more generally accurate approximations, one of which is

$$\widetilde{\kappa}_s(p_i) = \frac{3\,\big(\,\widetilde{\kappa}(p_{i+1}) - \widetilde{\kappa}(p_{i-1})\,\big)}{2\,a + 2\,b + d + e}\,, \tag{8}$$

where a, b are in (5), while

$$d = \| p_{i-1} - p_{i-2} \|, \qquad e = \| p_{i+2} - p_{i+1} \|. \tag{9}$$

In our application, since we have sampled the edges approximately uniformly, the two formulas (7), (8), did not lead to any observable difference in the efficacy of our algorithms.

Finally, we use the Euclidean invariant formula found by Boutin, [2], to approximate the torsion τ at p_i. The points $p_{i-1}, p_i, p_{i+1}, p_{i+2} \in C$ form a tetrahedron with base determined by p_{i-1}, p_i, p_{i+1}. The area of the base Δ is determined by using Heron's formula (6), while its volume is given by the vector triple product formula:

$$V = \frac{|(p_{i+2} - p_{i-1}) \cdot (p_{i+2} - p_i) \times (p_{i+2} - p_{i+1})|}{6}\,, \qquad \text{while} \qquad H = \frac{3\,V}{\Delta} \tag{10}$$

is its height. The invariant numerical approximation to torsion at the point p_i is

$$\widetilde{\tau}(p_i) = \frac{6\,H}{\widetilde{\kappa}(p_i)\,\| p_{i+2} - p_{i-1} \|\,\| p_{i+2} - p_i \|\,\| p_{i+2} - p_{i+1} \|}\,, \tag{11}$$

where $\widetilde{\kappa}(p_i)$ denotes the approximate curvature (4). One might object that formula (11) introduces an asymmetry with respect to the sample points centered at p_i. This can easily be corrected by averaging with the corresponding formula based on the sample points $p_{i-2}, p_{i-1}, p_i, p_{i+1} \in C$. However, as above, this more complicated formula does not lead to any increase in accuracy of our results. In light of these observations, we will employ the simpler formulae (4), (7), (11) to numerically approximate the Euclidean signature curve of our sampled space curve in a Euclidean invariant manner.

3.3. **Edge Matching.** Of course, when attempting to match the boundaries of two puzzle pieces, one should only compare suitable parts of their respective signatures, namely those corresponding to the a priori unknown edge that they have in common. This motivates dividing up their respective signatures into appropriate "signature codons". The method proposed in [14] is to partition the curve into *bivertex arcs*. By definition, a bivertex arc $B \subset C$ is a part of the curve lying between two successive *vertices*, defined as points where $\kappa_s = 0$. In other words, $\kappa_s \neq 0$ on all of B except at its two endpoints. The only parts of a plane curve not included in a bivertex arc are straight line segments and circular arcs, if any. Such "generalized vertices" are ignored by the puzzle solving algorithm. In practical terms, when dealing with noisy data, one determines when κ_s switches sign by introducing a threshold $\delta_0 > 0$ that needs to be crossed in order to fix an endpoint of a bivertex arc, [15]. One can then compare the individual bivertex arc signatures, which is supplemented by a procedure to reconstruct the rigid motion required to match several equivalent arcs on a candidate pair of puzzle pieces in order to ascertain whether or not they match.

We will similarly split a space curve representing a puzzle piece edge into bivertex arcs, using the same definition. We will determine whether two edges are a match by calculating a similarity score between their corresponding bivertex arcs. Observe that the excluded parts of the curve, where $\kappa_s \equiv 0$, are space curves of constant curvature. These include straight lines, circular arcs, and helices, which also have constant torsion. However, there exists a much larger variety of space curves that have constant curvature and non-constant torsion — see, for example, [25] — and our algorithm does not consider any such segments of the puzzle piece edges. In all the puzzles we have considered to date, this does not appear to be a significant weakness in our approach; nevertheless, further investigation into this aspect of the algorithm is warranted.

The similarity measure used to compare two bivertex arc signatures Σ_1 and Σ_2 will be based on the distance from the origin and angular position of each point thereon. For convenience, we will convert each signature point from Cartesian coordinates (κ, κ_s, τ) to spherical coordinates, which we denote by (ρ, θ, φ), so that

$$\rho = \sqrt{\kappa^2 + \kappa_s^2 + \tau^2}, \qquad \theta = \arctan\left(\frac{\kappa_s}{\kappa}\right), \qquad \varphi = \arccos\left(\frac{\tau}{\rho}\right). \tag{12}$$

Using a method introduced in [11], the comparison between these two sets of points will be based on the skewness measure of the cumulative magnitudes of their spherical coordinate representations.

Definition 1. Let $(r_1, \ldots, r_m)$, with $r_j \geq 0$, be an ordered of m-tuple non-negative real numbers. Their *cumulative magnitude* $\mathbf{R} = (R_1, \ldots, R_m)$ is defined recursively by

$$R_1 = r_1, \qquad R_k = r_k + \sum_{i=1}^{k-1} R_i = \sum_{i=1}^{k} (k - i + 1)\, r_i. \tag{13}$$

Given two discretized bivertex arc signatures, Σ^α containing m points and Σ^β containing n points, we will compute the respective cumulative distance magnitudes

$$\mathbf{R}^\alpha_\rho = (R^\alpha_{\rho,1}, \ldots, R^\alpha_{\rho,m}), \qquad \mathbf{R}^\beta_\rho = (R^\beta_{\rho,1}, \ldots, R^\beta_{\rho,n}),$$

based on their spherical radii ρ^α_j and ρ^β_j. To avoid potential anomalies, we will reorder the distributions so that the point with the middle index is the initial point and each successive point alternates between the left and right side of the initial point. The resulting cumulative

distance magnitudes are then compiled into a vector by negating those coming from the first bivertex arc:

$$\mathbf{R}_{\rho}^{\alpha,\beta} = (R_{\rho,1}^{\alpha,\beta}, \dots, R_{\rho,m+n}^{\alpha,\beta}) = (-R_{\rho,1}^{\alpha}, \dots, -R_{\rho,m}^{\alpha}, R_{\rho,1}^{\beta}, \dots, R_{\rho,n}^{\beta}), \tag{14}$$

The similarity between the two bivertex arcs is then quantified by the *skewness* of (14), cf. [16, p. 123], defined by

$$\Gamma_{\rho}^{\alpha,\beta} = \frac{\dfrac{1}{m+n}\displaystyle\sum_{k=1}^{m+n}(R_{\rho,k}^{\alpha,\beta})^3}{\left(\dfrac{1}{m+n}\displaystyle\sum_{k=1}^{m+n}(R_{\rho,k}^{\alpha,\beta})^2\right)^{3/2}}. \tag{15}$$

We can similarly determine the *longitudinal skewness* $\Gamma_{\theta}^{\alpha,\beta}$ and *latitudinal skewness* $\Gamma_{\varphi}^{\alpha,\beta}$ by adapting equations (13), (14), (15), to the azimuthal and zenith angle coordinates, respectively, taking the $2\,\pi$ periodicity of the latter into account. The overall *similarity score* between the two bivertex signatures is determined by

$$\Gamma^{\alpha,\beta} = |\Gamma_{\rho}^{\alpha,\beta}| + |\Gamma_{\varphi}^{\alpha,\beta}| + |\Gamma_{\theta}^{\alpha,\beta}|. \tag{16}$$

The similarity score's ability to detect matching edges is one of the most crucial components of our puzzle solving algorithm. The algorithm proposed in [15] is based on a computed "gravitational attraction" between the two bivertex edges, while the most conventional method for measuring similarity between two point sets is the Hausdorff distance, [26]. Our similarity measure works well for the problems at hand. It also has an interesting applications in the detection of similar and congruent curves. For example, if $\Gamma_{\rho}^{\alpha,\beta} = \Gamma_{\varphi}^{\alpha,\beta} = \Gamma_{\theta}^{\alpha,\beta} = 0$, then the two curves are congruent. On the other hand, two curves satisfying $\Gamma_{\varphi}^{\alpha,\beta} = \Gamma_{\theta}^{\alpha,\beta} = 0$ but $\Gamma_{\rho}^{\alpha,\beta} = c \neq 0$ are similar under a uniform scaling because the angular positions of the curves are equivalent.

4. Puzzle Assembly

Once the similarity scores between all bivertex arcs have been calculated, we must distinguish between false positives and true matches. We will consider potential matches as those whose similarity score (16) is less than a prescribed threshold, namely,

$$\Gamma^{\alpha,\beta} < \delta = 0.05.$$

We analyze the pairs of bivertex arcs in order, starting with the pair with the smallest score. Thus, given signatures corresponding to a pair of bivertex arcs, denoted by C^{α} and C^{β}, we seek to determine the rigid motion $g \in \mathrm{SE}(3)$ that will lead to optimal overlap.

The translation component of the sought for rigid motion will be taken as the displacement between the two centroids of the arcs. We can thus assume that both arcs have both been translated so that their centroids are at the origin. We then must determine the rotation matrix that optimally matches them. According to Schönemann, [34], given a pair of $n \times k$ matrices A, B, the nearest orthogonal matrix R which "roughly maps" A to B is given by $R = U\,V^T$, where U, V are the multiplicands in the singular value decomposition

$$M = A^T B = U\,\Sigma\,V^T. \tag{17}$$

In our case, we set A to be the $m \times 3$ matrix whose columns are the points in the first bivertex arc C^α, and B the $n \times 3$ matrix whose columns are the points in C^β. In order that M be a square matrix, the two bivertex arcs must contain the same number of points: $n = m$. If this is not the case, we smoothly spline interpolate the boundary points on the bivertex arc with fewer points and then randomly sample from the resulting spline in order to supplement the existing sample points. Using the resulting singular value decomposition formula (17) we are able to construct a rotation R matrix that best aligns the two bivertex arcs.

For each pair, we will determine whether the two pieces actually match by assembling them using a Procrustes algorithm based on the goodness of the fit provided by the surface area of the gap and overlap between the two potentially matched piece edges, say E^α and E^β. Each edge is resampled using a cubic spline so that each point on E^α is approximately aligned with a point on E^β with respect to the latitude. We will calculate the approximate surface area of the resulting spherical patch representing the combined gap and overlap between the two matched edges by modifying a composite trapezoidal rule.

In anticipation of our eventual application to (approximately) ellipsoidal egg shells, we will calculate the geodesic distance $d(p, q)$ between two points p and q using *Vincenty's formula*, [39]:

$$d(p,q) = 2 \arcsin \left(\frac{\sqrt{(\cos\varphi_q \sin\Delta\theta)^2 + (\cos\varphi_p \sin\varphi_q - \sin\varphi_p \cos\varphi_q \cos\Delta\theta)^2}}{\sin\varphi_p \sin\varphi_q + \cos\varphi_p \cos\varphi_q \cos\Delta\theta} \right). \tag{18}$$

Here, as above, θ and φ are the spherical angles (12), while

$$\Delta\theta = |\theta_p - \theta_q|.$$

For the n points on each sampled edge, the numerical quadrature formula for surface area is constructed by modifying the trapezoidal rule for the surface of a sphere:

$$A = \frac{1}{2} \sum_{i=1}^{[n/2]} d(p_{i-1}^{\alpha,*}, p_{i+1}^{\beta,*}) \left[d(p_{2i-1}^{\alpha}, p_{2i-1}^{\beta}) + d(p_{2i+1}^{\alpha}, p_{2i+1}^{\beta}) \right], \tag{19}$$

where the points $p_{i-1}^{\alpha,*}$ and $p_{i+1}^{\beta,*}$ are aligned to have the same longitude in order that $d(p_{i-1}^{\alpha,*}, p_{i+1}^{\beta,*})$ is the height of each trapezoid. If the computed surface area is larger than a preassigned threshold, specifically,

$$A > \varepsilon = 0.1,$$

then the match is deemed a false positive.

We also must verify that our potential match is between an outdented and an indented edge. We can determine the orientation of an edge by "removing" the edge from the piece and comparing the approximate surface area of the piece before and after the removal. Prior to removing the edge, we must calculate the approximate surface area A_1 by using our numerical quadrature formula (19). Next, we remove the edge by replacing it with a great circular arc between the initial and final point of the edge, then calculate the surface area A_2. If $A_1 > A_2$, then the edge is outdented, while $A_1 < A_2$ indicates an indented edge. If E^α and E^β are both outdented or indented, then the match is also identified as a false positive. We find that this approach suffices for most of the synthetically generated puzzles we considered, even in the presence of noise. However, in order to handle real-world scanned puzzle data, we anticipate that a more general procedure based on the piece locking algorithm of [15] should be implemented in future.

Once a match has been identified, we then perform the translation and rotation mappings on the two pieces as a whole, which generates a "matched pair" of puzzle pieces. We treat the result as a single puzzle piece, with the matching bivertex arcs removed. The process is then iterated using the resulting smaller collection of pieces. The algorithm terminates either with a successful reassembly of the puzzle, or when no further piece matches satisfying our goodness of fit criterion.

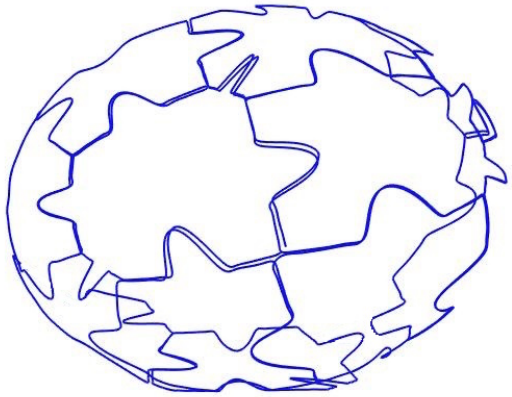

FIGURE 4. Reassembled Synthetic Puzzle

When the matching algorithm is completely successful, a totally reassembled puzzle will be obtained. In Figure 4, we display an example of a synthetically generated spherical puzzle containing 18 pieces that has been solved using our algorithm. Our three-dimensional puzzle solving algorithm has successfully solved synthetic jigsaw puzzles containing up to sixty pieces.

Although reassembling synthetic puzzles is certainly of interest, and serves to demonstrate proof of concept, our guiding aim is to be able to reassemble broken objects scanned into a computer. In order to simulate actual data, we introduce noise by applying a random motion to each sample point on the piece. Explicitly, for a given point p, we add noise to produce the point $\tilde{p} = (1 + \nu\, t)\, p$, where ν is the noise threshold and $0 < t \leq 1$ is a uniformly distributed random number. Our MATLAB programs have successfully reassembled noisy puzzles that contain up to twenty pieces with threshold $\nu \leq 0.02$. As expected, reassembling noisy puzzles is significantly more difficult. One of the greatest challenges is to obtain a precise segmentation so that matching edges have approximately the same initial and final points. Indeed, since our similarity score is recursively defined, misaligned initial and final points of two matching edges can produce a higher similarity score, and thus erroneously flagged as a false match. We have not yet investigated how to recover from such mistaken matches through some form of backtracking algorithm.

5. Assembly of a Broken Egg Shell

Having successfully tested our algorithm on artificially generated data, we turn our attention to some scanned data of a broken ostrich egg that was supplied to us by Marshall Bern, [1], in response to the jigsaw puzzle assembly paper of Hoff and the third author, [15]. Bern's data was scanned by Nina Amenta using a turntable, a stripe of light, and a video camera, and consists of 15 distinct pieces subject to unknown rigid motions from their

original assembled configuration. Each data file is in `.off` format, and includes a list of approximately 30,000 data points obtained from sampling the egg shell surface.

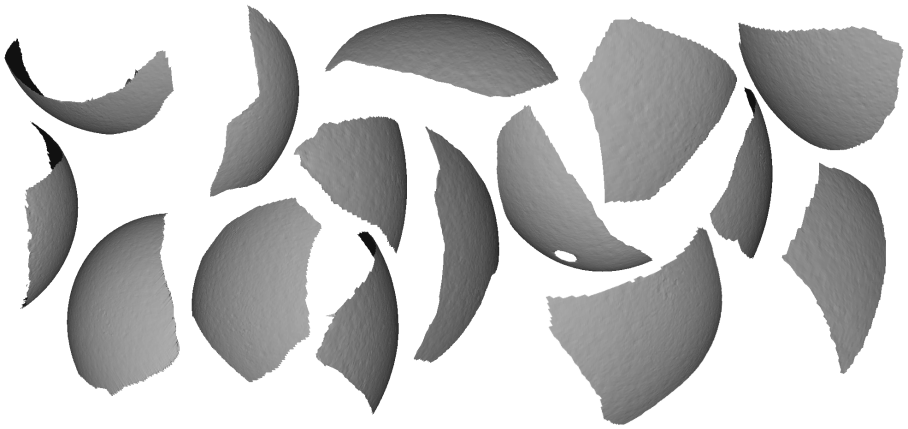

FIGURE 5. The Ostrich Egg Pieces

In our investigations of the reassembly of the egg shell, we have operated under the assumption that, at the time of scanning, some much smaller pieces, "chips," were not included. This makes finding legitimate piece matches more challenging, as these chips may fit between one or more pieces, leaving gaps in the assembled egg shell. To accommodate these missing pieces, additional spline-based smoothing was required.

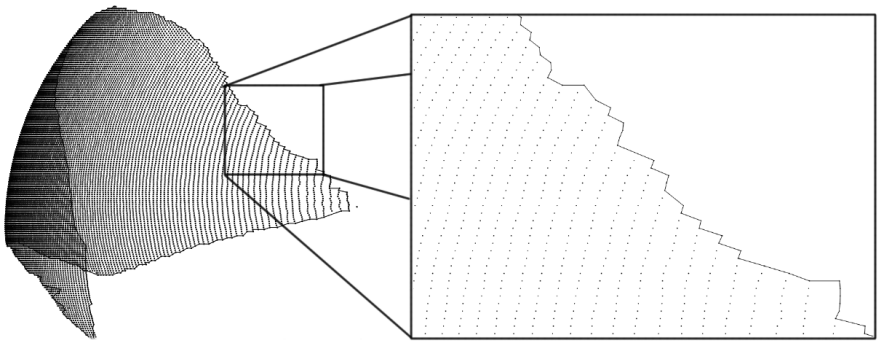

FIGURE 6. The Boundary of an Egg Piece

The first step is to extract the boundaries of individual egg shell pieces. All of our signature methods are dependent on having an *ordered* set of points to be used as a boundary. In the data supplied, 12 of the individual piece files contained ordered boundary data, which we used in what follows. As for the remaining 3 files, we applied an "expanding snake" method, [5, 17, 18], adapted to the point cloud data representing the surface of the piece, to extract the requisite ordered boundary; we obtained usable ordered boundaries in all but 1 of the remaining pieces, and this problematic piece was subsequently ignored.

The next step is to smooth each piece's boundary in order to mitigate the effect of jaggedness, missing tiny chips, and other artifacts. We employed a periodic smoothing spline to smooth out these jagged boundaries, and, hopefully, obtain an approximation of the egg

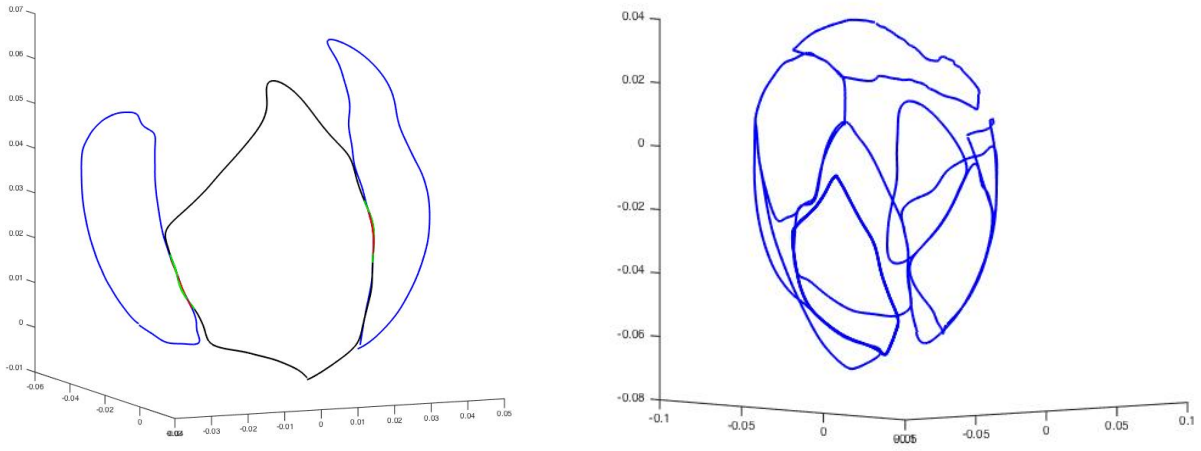

FIGURE 7. Partially Assembled Ostrich Egg — 3 and 11 pieces

shell piece boundaries as if no chip was missing. However, this method is only able to handle relatively small missing chips, and does not easily extend to larger omissions.

Automatic reassembly of the broken ostrich egg proved to be challenging, in particular because of the difficulty in automatically verifying the authenticity of matches on such noisy data. In particular, reassembling the egg presented several problems not present in the synthetic puzzles. Most notable was the appearance of false matches. A substantial number of "matching" signature curves were almost certainly wrong, as they failed to maintain the expected surface curvature of the egg shell. As a result, each potential match produced by the signature curve algorithm needed to be verified with a visual check. In the end, by applying this combination of automatic and visual checking, we were able to generate a partial reassembly of 11 pieces of the egg shell data, as depicted in Figure 7.

6. Conclusion and Further Work

We have described the initial phases in the design of an algorithm for automatically assembling apictorial three-dimensional jigsaw puzzles obtained by breaking a smooth surface into pieces. In order to run preliminary tests on our methods, we developed a new algorithm for constructing such puzzles from spheres and ellipsoids, with the methods being, in principle, applicable to fairly general surfaces in $\mathbb{R}^3$. An important next phase of the project will be to adapt our Voronoi-based methods to generate puzzles on more general curved surfaces, including convex, non-convex, and even those with nontrivial topology such as tori. While the generation of puzzles will require some additional work, we do not anticipate our puzzle reassembly algorithm to experience any new difficulties in the reconstruction process since it only relies on the shapes of the individual pieces and does not require any a priori knowledge of the final reconstructed surface.

Our solution to the space curve matching problem relies on the use of differential invariant signatures, an approach that, based on the modern method of equivariant moving frames has seen significant recent developments in both theoretical and practical directions, [30]. In our case, curvature, torsion, and the derivative of curvature with respect to arc

length are employed to parametrize the Euclidean signature curve that characterizes space curves uniquely up to rigid motion. Our comparison of puzzle piece edges relies on matching a pair of bivertex arcs, which is based on comparison of their respective signature curves. This method is quite robust when dealing with the synthetic jigsaw puzzles generated by our algorithm, even in the presence of noise. One issue worth further investigation is whether the bivertex arcs are the optimal signature codons that serve to semi-localize the differential invariant signature comparison. Preliminary tests indicate that choosing arcs bounded by zeros of curvature, $\kappa = 0$, lead to a more robust and accurate matching algorithm. Moreover, in contrast to the large number of constant curvature space curves, for which $\kappa_s \equiv 0$ and which our current algorithm cannot handle, the only problematic curves for which $\kappa \equiv 0$ are straight line segments. (On the other hand, torsion is undefined at such inflection points.)

So far, our algorithms have been only moderately successively on the one set of real-world data we have tested them on — the broken ostrich eggshell. The next phase of the project will be to generate an extensive collection of scanned data of real-world three-dimensional puzzles obtained from broken surface objects with a variety of shapes, e.g. egg shells, ceramics, glassware, etc., and then to further refine the smoothing and signature matching algorithms as necessary. Indeed, to ensure robustness when dealing with real-world data, a more sophisticated approach will no doubt be required, and the most promising would be an adaptation of the planar piece locking algorithm in [15], that relies on matching several bivertex arcs. With this in hand, we will be in a position to directly compare our methods with existing alternatives, e.g. [38].

Real-world applications, in particular to broken archaeological objects such as pottery fragments, [21], as well as art restoration, [22], are directions eminently worth further study. Moreover, adding pictorial information to our current purely shape-based approach will, we anticipate, lead to significant advances on these challenging and important problems.

Acknowledgments: We would like to thank Marshall Bern for sending us the broken ostrich egg data that inspired this work. The work of the first, third, and fifth authors was supported in part by NSF grant DMS–1108894. The work of the sixth author was supported in part by NSF grant DMS–0839966.

References

[1] Bern, M., personal communication, 2013.

[2] Boutin, M., Numerically invariant signature curves, *Int. J. Computer Vision* **40** (2000), 235–248.

[3] Calabi, E., Olver, P.J., Shakiban, C., Tannenbaum, A., and Haker, S., Differential and numerically invariant signature curves applied to object recognition, *Int. J. Computer Vision* **26** (1998), 107–135.

[4] Cartan, É., Les problèmes d'équivalence, *in*: *Oeuvres Complètes*, part. II, vol. 2, Gauthier–Villars, Paris, 1953, pp. 1311–1334.

[5] Caselles, V., Kimmel, R., and Sapiro, G., Geodesic active contours, *Int. J. Computer Vision* **22** (1997), 61–79.

[6] Chen, J., Zhao, X., and Li, Z., An algorithm for the generation of Voronoi diagrams on the sphere based on QTM, *Photogrammetric Engin. Remote Sensing* **69** (2003), 79–89.

[7] Chung, F.R.K., *Spectral Graph Theory*, CBMS Regional Conference Series in Mathematics, No. 92, Amer. Math. Soc., Providence, R.I., 1997.

[8] Coifman, R.R., Lafon, S., Lee, A.B., Maggioni, M., Nadler, B., Warner, F., and Zucker, S.W., Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps, *Proc. Nat. Acad. Sci.* **102** (2005), 7426–7431.

[9] Craciun, G., Garcia–Puente, L.D., and Sottile, F., Some geometrical aspects of control point for toric patches, *in*: *Mathematical Methods for Curves and Surfaces*, M. Dæhlen et. al., eds., Lecture Notes in Computer Science, vol. 5862, Springer–Verlag, New York, 2010, pp. 111–135.

[10] Goldberg, D., Malon, C., and Bern, M., A global approach to the automatic solution of jigsaw puzzles, *Computational Geometry* **28** (2004), 165–174.

[11] Grim, A., and Shakiban, C., Applications of signatures in diagnosing breast cancer, *Minnesota J. Undergrad. Math.* **1** (1) (2015), 001.

[12] Guggenheimer, H.W., *Differential Geometry*, McGraw–Hill, New York, 1963.

[13] Habicht, W., and van der Waerden, B.L., Lagerung von Punkten auf der Kugel, *Math. Ann.* **123** (1951), 223–234.

[14] Hoff, D., and Olver, P.J., Extensions of invariant signatures for object recognition, *J. Math. Imaging Vision* **45** (2013), 176–185.

[15] Hoff, D., and Olver, P.J., Automatic solution of jigsaw puzzles, *J. Math. Imaging Vision* **49** (2014), 234–250.

[16] Hogg, R.V., Tanis, E.A., and Zimmerman, D.L., *Probability and Statistical Inference*, 5th ed., Prentice–Hall, Inc., Upper Saddle River, N.J., 1997.

[17] Kass, M., Witkin, A., and Terzopoulos, D., Snakes: active contour models, *Int. J. Computer Vision* **1** (1988), 321–331.

[18] Kichenassamy, S., Kumar, A., Olver, P.J., Tannenbaum, A., and Yezzi, A., Gradient flows and geometric active contour models, *in*: *Fifth International Conference on Computer Vision*, IEEE Computer Soc. Press, Cambridge, Mass., 1995, pp. 810–815.

[19] Kishon, E., and Wolfson, H., 3-D curve matching, Technical Report No. 283, Courant Institute, New York University, 1987.

[20] Kong, W., and Kimia, B.B., On solving 2D and 3D puzzles using curve matching, *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2** (2001), 583–590.

[21] Leitão, H.C.G., and Stolfi, J., A multiscale method for the reassembly of two-dimensional fragmented objects, *IEEE Trans. Pattern Analysis Machine Intelligence* **24** (2002), 1239–1251.

[22] Leutwyler, K., Solving a digital jigsaw puzzle, *Scientific American*; June 25, 2001, `http://www.scientificamerican.com/article/solving-a-digital-jigsaw/`

[23] Lubotsky, A., Phillips, R., and Sarnak, P., Hecke operators and distributing points on the sphere I, *Commun. Pure Appl. Math.* **39** (1986), 149–186.

[24] Luo, C., Ge, X., and Wang, Y., Uniformization and density adaptation for point cloud data via graph Laplacian, Tech report OSU-CISRC-11/14-TR19, Ohio State University, 2014.

[25] Monterde, J., Salkowski curves revisited: A family of curves with constant curvature and non-constant torsion, *Computer Aided Geometric Design* **26** (2009), 271–278.

[26] Mumford, D., The problem of robust shape descriptors, *in*: *Proceedings of the First International Conference in Computer Vision*, IEEE Comput. Soc. Press, Washington DC, 1987, pp. 602–606.

[27] Newman, M., *Networks: An Introduction*, Oxford University Press, Oxford, 2010.
[28] Okabe, A., Boots, B., Sugihara, K., and Chiu, S.N., *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, 2nd ed., John Wiley & Sons, New York, 2000.
[29] Olver, P.J., *Equivalence, Invariants, and Symmetry*, Cambridge University Press, Cambridge, 1995.
[30] Olver, P.J., Lectures on moving frames, *in*: *Symmetries and Integrability of Difference Equations*, D. Levi, P. Olver, Z. Thomova, and P. Winternitz, eds., London Math. Soc. Lecture Note Series, vol. 381, Cambridge University Press, Cambridge, 2011, pp. 207–246.
[31] Rakhmanov, E.A., Saff, E.B., and Zhou, Y.M., Minimal discrete energy on the sphere, *Math. Res. Lett.* **1** (1994), 647–662.
[32] Saff, E.B., and Kuijlaars, A.B.J., Distributing many points on a sphere, *Math. Intelligencer* **19** (1997), 5–11.
[33] Sağiroğlu, M.S., and Erçil, A., A texture based approach to reconstruction of archaeological finds, *in*: *Proceedings of the 6th International conference on Virtual Reality, Archaeology and Intelligent Cultural Heritage, VAST05*, Mudge, M., Ryan, N., Scopigno, R., eds., Eurographics Assoc., Aire-la-Ville, Switzerland, 2005, pp. 137–142.
[34] Schönemann, P., A generalized solution of the orthogonal Procrustes problem, *Psychometrika* **31** (1969), 1–10.
[35] Smale, S., Mathematical problems for the next century, *Math. Intelligencer* **20** (1998), 7–15.
[36] Spivak, M., *A Comprehensive Introduction to Differential Geometry*, vol. 2, 2nd ed., Publish or Perish, Wilmington, Delaware, 1979.
[37] Thomas, T.P., Anderson, D.D., Willis, A.R., Liu, P., Frank, M.C., Marsh, J.L., and Brown, T.D., A computational/experimental platform for investigating three-dimensional puzzle solving of comminuted articular fractures, *Comput. Methods Biomech. Biomed. Engin.* **14** (2011), 263–270.
[38] Üçoluk, G., and Toroslu, I.H., Reconstruction of broken surface objects, *Computers & Graphics* **23** (1999), 573–582.
[39] Vincenty, T., Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations, *Survey Rev.* **23** (1976), 88–93.
[40] Wolfson, H., Schonberg, E., Kalvin, A., and Lamdan, Y., Solving jigsaw puzzles by computer, *Ann. Operations Research* **12** (1988), 51–64.
[41] Yao, F.–H., and Shao, G.–F., A shape and image merging technique to solve jigsaw puzzles, *Pattern Recognition Lett.* **24** (2003), 1819–1835.
[42] Yates, C., The Baffler: the Nonagon, Ceaco, Newton, MA, 2010.

Department of Mathematics, University of St. Thomas, St. Paul, MN 55105-1096, USA
E-mail address: grim4684@stthomas.edu

Department of Economics, University of Oxford, Oxford, UK
E-mail address: timothy.oconnor@nuffield.ox.ac.uk

School of Mathematics, University of Minnesota, Minneapolis, MN 55455, USA
E-mail address: olver@umn.edu
URL: http://www.math.umn.edu/~olver

Department of Mathematics, University of St. Thomas, St. Paul, MN 55105-1096, USA
E-mail address: cshakiban@stthomas.edu
URL: http://courseweb.stthomas.edu/cshakiban/

Department of Mathematics, University of St. Thomas, St. Paul, MN 55105-1096, USA
E-mail address: ryan@stthomas.edu

Department of Mathematics and Statistics, Carleton College, Northfield, MN 55057, USA
E-mail address: rthompson@carleton.edu
URL: http://people.carleton.edu/~rthompson/