Voronoi Scissors: Approximate Dissection of 2D Shapes with Continuous Optimization

ANRAN QI, The University of Tokyo, Japan and Centre Inria d'Université Côte d'Azur, France
NICO PIETRONI, University of Technology Sydney, Australia
MIKHAIL BESSMELTSEV, Université de Montréal, Canada
NOBUYUKI UMETANI, The University of Tokyo, Japan
ADRIEN BOUSSEAU, Centre Inria d'Université Côte d'Azur, France
TAKEO IGARASHI, The University of Tokyo, Japan

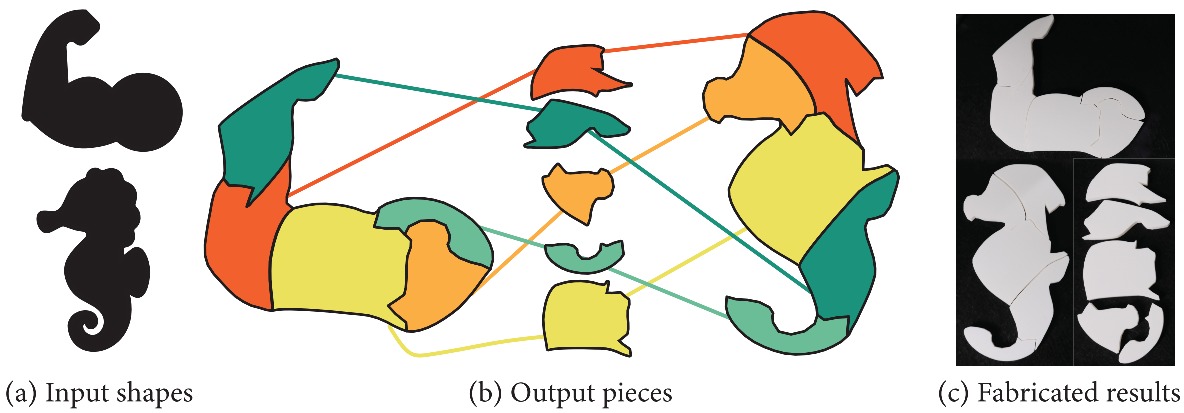


Fig. 1. Our algorithm takes two shapes as input (a) and computes an approximate dissection of these shapes into a common set of pieces (b). Each shape can be reproduced by rearranging the pieces via rigid transformations (c, 3D-printed pieces).

The dissection of planar shapes into a common set of reusable pieces is a classical problem, with connections to geometry processing, fabrication, and design. Despite its long-standing interest, the problem remains challenging due to the interplay of geometric constraints and combinatorial complexity. Existing approaches are limited to exact dissection of simple polygonal shapes, or can only handle a small number of pieces and introduce significant distortion when performing approximate dissection of more complex shapes.

We present a fully automatic method for dissecting two-dimensional shapes into a shared set of pieces that can be rearranged to approximate multiple target configurations. Our approach formulates the problem as a hybrid optimization pipeline. We first perform a continuous optimization based on a Voronoi-driven decomposition to generate an initial partition. This is followed by a global optimization step that determines a consistent rearrangement of pieces while enforcing structural constraints required for valid, rigid, and non-overlapping assemblies. The final refinement stage refines the solution to best approximate the input shapes.

Our method guarantees that all pieces admit rigid, non-overlapping rearrangements. Moreover, our pieces are more compact and yield less distortion than the ones obtained by prior work. We demonstrate the generality of our approach across novel dissection scenarios, including multi-shape rearrangements, shapes with complex topology, and reuse-driven settings where a fixed source shape is partitioned to approximate a target.

CCS Concepts: • **Computing methodologies** → **Shape modeling**; **Shape analysis**.

Additional Key Words and Phrases: Dissection, Voronoi diagrams

ACM Reference Format:
Anran Qi, Nico Pietroni, Mikhail Bessmeltsev, Nobuyuki Umetani, Adrien Bousseau, and Takeo Igarashi. 2018. Voronoi Scissors: Approximate Dissection of 2D Shapes with Continuous Optimization. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, Article 111, 15 pages. https://doi.org/XXXXXXX.XXXXXXX

Authors' Contact Information: Anran Qi, The University of Tokyo, Japan and Centre Inria d'Université Côte d'Azur, France; Nico Pietroni, University of Technology Sydney, Australia; Mikhail Bessmeltsev, Université de Montréal, Canada; Nobuyuki Umetani, The University of Tokyo, Japan; Adrien Bousseau, Centre Inria d'Université Côte d'Azur, France; Takeo Igarashi, takeo@acm.org, The University of Tokyo, Japan.

1 Introduction

A geometric dissection of two shapes of equal area is a partitioning of these shapes into the same set of pieces, allowing any of the shapes to be transformed into the other by rearranging the pieces through rigid transformations (Fig. 1). Geometric dissection has fascinated mathematicians for centuries, ranging from millennia-old visual proofs of Pythagoras' Theorem [Maor 2007] to the very recent proof that Dudeney's dissection of the square and the equilateral triangle is optimal in terms of the number of pieces [Demaine et al. 2026].

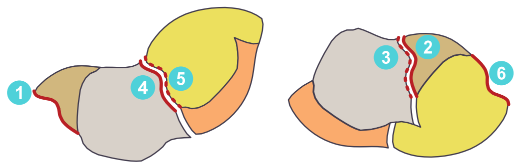


Fig. 2. A valid dissection needs to satisfy long chains of constraints on the boundary segments of multiple pieces. In this example, segment 1 corresponds to segment 2, which must match segment 3. Segment 3 corresponds to segment 4, which in turn matches segment 5. Finally, segment 5 corresponds to segment 6. The chain begins and ends at segments that lie on the silhouettes of the shapes.

In addition to being an intriguing recreational puzzle, geometric dissection holds practical value for sustainable design, as it allows the creation of multiple shapes from a fixed set of parts [Song et al. 2017], or the reuse of parts of an existing shape to create a new, different shape [Baas et al. 2025].

While any pair of polygons of equal area can be dissected into a finite number of pieces, finding whether two polygons can be dissected into a given number K of pieces is NP-hard [Bosboom et al. 2015]. This difficulty motivated Duncan et al. [2017] to introduce the problem of *approximate* dissection, where the goal is to find a set of K pieces that can be rigidly transformed to reproduce two shapes *as best as possible*, as measured by a suitable distance between the pieces' arrangements and the target shapes. We adopt the same problem definition and propose an automated algorithm that finds dissections with reduced approximation error than those shown by Duncan et al. [2017], while producing more compact pieces.

Despite the additional freedom offered by this problem setting, finding an approximate K-dissection of two arbitrary shapes remains difficult because the joint partitioning of two shapes yields a vast design space to explore, in which the shape of the pieces and their rigid transformations are free variables. In addition, adjacent pieces need to match along their shared boundary, without any gaps or overlaps. A key challenge is that correspondences along the shared boundary induce long-range dependency chains across multiple pieces (Fig. 2). Small inconsistencies can therefore propagate globally, making it difficult to construct valid rigid assemblies.

Duncan et al. [2017] address these challenges by identifying pairwise correspondences between boundary segments of the two shapes, and enumerating candidate dissections formed by connecting these correspondences using a tree search. But the combinatorial nature of the tree search prevents it from scaling to a large number of pieces. Moreover, during combinatorial optimization, the method employs a turning-angle–based error metric for fast pruning, while ignoring length differences. In practice, this often yields complex pieces with fragile, elongated parts. The key strength of our work is to formulate the joint partitioning of two shapes as a *continuous* optimization of the piece shapes using a differentiable Voronoi diagram. This formulation enables direct control over the resulting piece shapes through the design of loss functions, while avoiding the limitations of the *combinatorial* approach of Duncan et al. [2017].

However, the connectivity of the pieces evolves continuously through Voronoi optimization, which prevents us from detecting and imposing chains of hard constraints across their shared boundary segments. We impose these constraints in a second step, once the connectivity of the pieces has converged. Specifically, we first simplify the pieces into coarse polygons, which we then deform to be identical in both arrangements while preserving the connectivity found by the Voronoi optimization. We then refine the boundaries of these polygons to best approximate the input shapes while keeping the pieces rigid.

We demonstrate our method by dissecting diverse pairs of shapes, including some shown in prior work for comparison. In addition to achieving higher accuracy and producing more compact pieces than the algorithm of Duncan et al. [2017], our algorithm can handle a larger number of pieces, extend to shapes composed of multiple disconnected components, to shapes containing holes, and to reuse scenarios where one of the two shapes is reproduced exactly.

2 Related Work

The dissection of planar shapes into a set of reusable and recombinable pieces is a challenging problem that spans multiple interconnected optimization tasks, including the design of reconfigurable objects and the reuse of components from existing shapes in upcycling scenarios.

2.1 Computational Dissection Design

Introduced in the early 19th century, the Wallace–Bolyai–Gerwien theorem states that any two polygons of equal area admit a finite dissection into a common set of pieces [Bolyai 1832; Gerwien 1833]. Abbott et al. [2012] prove a similar result for the case of hinged dissections, where all pieces are connected in a chain by joints, enabling continuous transformations between configurations without disconnecting components. However, practical applications seek dissections with a minimum number of pieces, which is an NP-hard problem [Bosboom et al. 2015]. The complexity of geometric dissection has motivated numerous studies at the frontier of mathematics and puzzle design [Frederickson 2002; Pargeter 2004], but in most cases these studies focus on simple polygonal domains, such as rectilinear shapes [Czyzowicz et al. 2007] or specific geometric figures [Demaine et al. 2026].

Building on these foundations, computational methods have been introduced to handle more general inputs. Zhou and Wang [2012] propose a stochastic search framework over discrete grids to generate dissections for arbitrary 2D shapes, which was later extended to voxel grids for volumetric dissections [Tang et al. 2019]. Discrete grids also form the basis of the algorithm by Kita [2023] for dissecting shapes into colored polyominoes. Closer to our goal are the methods proposed by Duncan et al. [2017] and by Li et al. [2018b] for standard and hinged dissections respectively, where the input shapes are only *approximately* reproduced. Deviating from the input provides more freedom to find valid dissections, and allows us to optimize for complementary objectives, such as compactness of the pieces. Note that Li et al. [2018b] search in a large database of shapes for pairs that correspond well under a hinged dissection. In contrast, we aim at finding the best dissection for arbitrary pairs of shapes, which is why we compare our results to the ones produced by Duncan et al. [2017].

2.2 Modular Design and Reuse

A related line of work focuses on decomposing shapes into components that can be reused across different configurations. Applications of this paradigm range from reconfigurable furniture [Song et al. 2017] to garment reuse [Qi et al. 2025]. A broader overview of these approaches is provided in Wang et al. [2021].

An important application for modular design is architecture [Chen et al. 2023; Eigensatz et al. 2010; Fu et al. 2010; Singh and Schaefer 2010], where the benefits of reuse are particularly pronounced at scale. Recent work investigates the segmentation of wind turbine blades into reusable panels [Joustra et al. 2021; Pupping et al. 2025], as well as the assembly of truss and grid structures from reused or unconventional elements [Brütting et al. 2019; Colabella et al. 2017; Favilli et al. 2025; Kovacs et al. 2017] and the creation of functional fixtures from everyday objects [Mei et al. 2024]. Other approaches rely on predefined modular systems. For example, Zometool provides a construction paradigm based on a fixed set of struts of varying lengths connected through specialized nodes, enabling the assembly of a wide range of three-dimensional structures [Zimmer et al. 2014]. Similar strategies have also been explored in garment design, either to create multiple garments from the same modules [Del Valle et al. 2025; Lin et al. 2025], or to reuse existing garments to create new ones [Qi et al. 2025]. More related to shape dissection is the approach by Baas et al. [2025] that extracts a compact set of surface panels from one or more existing geometries and reassembles them to approximate a target shape via partial registration. However, the resulting panels do not connect seamlessly. Our approach does not assume predefined primitives or discrete element libraries and derives seamless partitions made of reusable elements directly from the input geometry.

2.3 Voronoi Diagrams in Graphics

Voronoi-based methods have a long-standing tradition in computer graphics, with their primary use in meshing and remeshing through centroidal Voronoi tessellations [Lévy and Liu 2010; Liu et al. 2009]. Beyond meshing, Voronoi structures have been successfully applied across a wide range of domains. For instance, they have been used in physical simulation [Busaryev et al. 2012; de Goes et al. 2015], in architectural geometry for structural optimization [Tonelli et al. 2016], and in topology optimization [Lu et al. 2014]. More recently, differentiable formulations of Voronoi diagrams have been used to address inverse problems in cellular mechanics [Feng et al. 2023; Numerow et al. 2024]. Recent developments further extend these ideas to differentiable rendering and image processing [Govindarajan et al. 2026; Iinbor et al. 2026]. We build our algorithm on the work of Wu et al. [2024], which optimizes Voronoi diagrams to generate partitions of 2D shapes for floor plan design.

In contrast to the above approaches that typically rely on Voronoi diagrams to decompose a single shape, our algorithm leverages Voronoi structures to jointly partition two shapes. The evolution of the Voronoi diagram is jointly optimized with a rigid transformation of the Voronoi cells, such that the pieces formed by agglomerated cells register across the two shapes. This formulation revisits the Voronoi partition as a correspondence-aware representation that couples partitioning and structural matching within a unified optimization framework.

3 Requirements and Overview

Our inputs are a pair of 2D shapes, A and B, possibly containing holes and/or disconnected components, and K, the number of dissection pieces. Each shape is represented as a set of polylines encoding exterior and interior boundaries. We seek to dissect the two shapes into two arrangements of the same K pieces, with the following requirements:

- Isometry. Each piece should have exactly the same shape in both arrangements, up to a rigid transformation.
- Seamless connection. In both arrangements, the pieces should connect seamlessly without overlaps nor gaps.
- Shape approximation. Both arrangements should approximate the corresponding input shapes as best as possible.

Satisfying all three requirements is challenging: In particular, it is unclear how to enforce a seamless connection between pieces when the connectivity between them is not known.

As shown in the Figure 3, our method satisfies the three requirements in three stages. We first relax the isometry requirement and partition the two shapes into K pairs of corresponding pieces so that the pieces in each pair are as close as possible up to a rigid transformation (Fig. 3b, Sec 4). After this initial stage, the input shapes are partitioned without overlaps or gaps, but the corresponding pieces may not be exactly the same.

In the second stage, we fix the connectivity of this initial partition and enforce the isometry requirement by relaxing the shape approximation (Fig. 3c, Sec 5.1). Precisely, we simplify the pieces into coarse polygons connected via shared vertices along their boundaries, and we deform these polygons to be identical in the two arrangements after rigid transformations.

In the final stage, we enforce all three requirements: We fix the rigid transformations, and, keeping the pieces connected, we optimize their boundaries to better align with the input shapes while enforcing that they remain identical in the two arrangements (Fig. 3d, Sec 5.2).

4 Voronoi-based Partition Initialization

The goal of our initial stage is to partition both shapes into pairs of nearly isometric pieces. Our main insight is to formulate this problem as a continuous optimization that, starting with a random initial partition, converges towards a partition that best satisfies our requirements. Since such a continuous optimization only finds local minima, we run multiple instances in parallel to explore the solution space.

We first sample $K \times M$ points inside both the input shapes A and B. Here, we denote these sets of points $\mathcal{P}^A = \{\mathbf{p}^A_{km}\}^{K,M}_{k=1,m=1}$ and $\mathcal{P}^B = \{\mathbf{p}^B_{km}\}^{K,M}_{k=1,m=1}$ where $\mathbf{p}^A_{km}, \mathbf{p}^B_{km} \in \mathbb{R}^2$. We use Poisson disk sampling as the initialization to cover the two shapes uniformly, and we adjusted the M such that the $K \times M$ is roughly $N = 125$ in our implementation. We then treat each point as the site of a cell in a Voronoi diagram and express each piece as the union of the Voronoi cells assigned to that piece. This process effectively

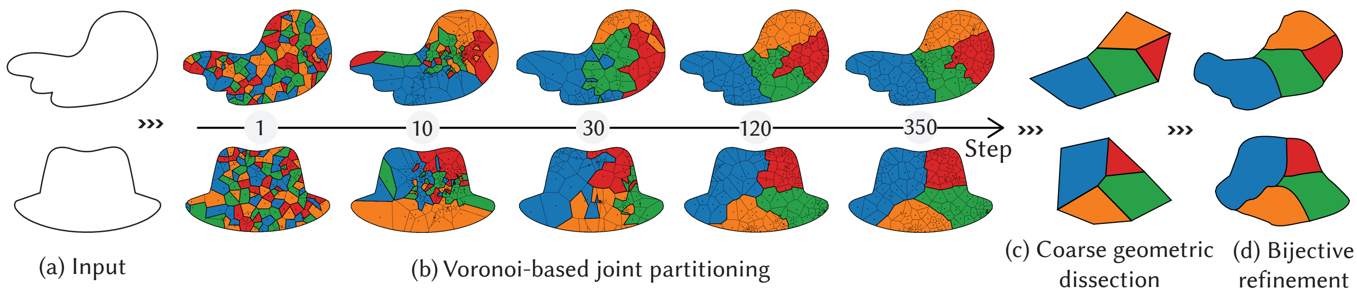


Fig. 3. Method Overview. Starting from two input shapes (a), we perform Voronoi-based joint partitioning to obtain an initial set of approximately isometric piece pairs (b), followed by coarse geometric dissection to enforce exact isometry (c), and bijective refinement to better approximate the input shapes (d).

produces a partition of the two shapes into K pieces made of — possibly disjoint — cells.

We next optimize the site positions so that the corresponding cells agglomerate into compact pieces that are as similar as possible in the two shapes when registered by a rigid transformation. We express this goal as the minimization of an objective function:

$$\arg\min_{\mathcal{P}^A,\mathcal{P}^B} w_{\text{iso}}\,\mathcal{L}_{\text{iso}} + w_{\text{area}}\,\mathcal{L}_{\text{area}} + w_{\text{balance}}\,\mathcal{L}_{\text{balance}} + \mathcal{L}_{\text{reg}}. \quad (1)$$

The main component of this function is $\mathcal{L}_{\text{iso}}$, which promotes the approximate isometry between the corresponding pieces (Sec. 4.1). The two terms $\mathcal{L}_{\text{area}}$ and $\mathcal{L}_{\text{balance}}$ control the areas of the Voronoi cells and pieces (Sec. 4.2). Finally, $\mathcal{L}_{\text{reg}}$ is a regularization that promotes the sites to be close to their Voronoi cell centroids (reminiscent of [Du et al. 1999]) and makes the boundaries of the Voronoi cells simple. In addition, the regularization aims to keep the pieces connected: Although each Voronoi cell is convex, the pieces may not be. Non-convex pieces therefore require multiple sites, yet the union of all Voronoi cells of a piece is not always a connected set (Fig. 3b). In practice, thanks to this regularization, each piece is connected by the end of the optimization. The complete regularization term follows closely Wu et al. [2024] and is given in the Suppl. 1 for completeness.

The weights in all our experiments are set to $w_{\text{iso}} = 0.5$, $w_{\text{area}} = 2$ and $w_{\text{balance}} = 10$. We minimize the energy via gradient descent with 350 steps.

4.1 Promoting isometry

To promote isometry, we perform rigid registration for each pair of corresponding pieces at each iteration of our optimization and measure the distance between the boundaries of the registered pieces.

First, we extract the boundaries of the corresponding pieces from their respective Voronoi diagrams. Since the pieces are not necessarily connected, we only consider the largest connected boundary component of each, as measured by enclosed area. With the index $k \in \{1, \ldots, K\}$ iterating over corresponding pieces in shapes A and B, we represent the piece boundaries as $\gamma_k^A = \{\mathbf{a}_1, \ldots, \mathbf{a}_{T_A}\}$ and $\gamma_k^B = \{\mathbf{b}_1, \ldots, \mathbf{b}_{T_B}\}$ counterclockwise loops.

We use the coherent point drift (CPD) algorithm [Myronenko and Song 2010] to compute the rotation $\mathbf{R}$ between boundary loops γ_k^A and γ_k^B. For piece k, we define the isometry loss as

$$\mathcal{L}_{\text{iso}}^{(k)} = \sum_{t_A=1}^{T_A} \sum_{t_B=1}^{T_B} W_{t_A t_B} \left\| \mathbf{R}(\mathbf{b}_{t_B} - \bar{\mathbf{b}}) - (\mathbf{a}_{t_A} - \bar{\mathbf{a}}) \right\|^2, \quad (2)$$

where $\bar{\mathbf{a}}$ and $\bar{\mathbf{b}}$ are the centroids of the corresponding loops, leveraging the fact that the optimal translation aligns centroids of two point sets [Myronenko and Song 2010]. Here $W_{t_A t_B}$ encodes the soft correspondence weights computed by CPD. Since the algorithm is sensitive to the initialization of $\mathbf{R}$, we try multiple initializations and choose the best result. The total isometry loss is then

$$\mathcal{L}_{\text{iso}} = \sum_{k=1}^{K} \mathcal{L}_{\text{iso}}^{(k)}. \quad (3)$$

The details of this step and the gradient of $\mathcal{L}_{\text{iso}}$ with respect to the site locations $\mathcal{P}^{\{A,B\}}$ are detailed in Suppl. 6.

4.2 Controlling the areas

We follow [Wu et al. 2024] to compute the area $S_{km} \geq 0$ of the m^{th} Voronoi cell of piece k. We then encourage the corresponding pieces in shapes A and B to have the same areas with the term

$$\mathcal{L}_{\text{area}} = \sum_{k=1}^{K} \left(\sum_{m=1}^{M} S_{km}^A - S_{km}^B \right)^2. \quad (4)$$

Since the pieces we search for are approximate rigid transformations of each other, this term is not strictly necessary, but we found it helps speed up the convergence (Fig. 8).

In addition, we prevent Voronoi cells from collapsing by minimizing the area variance within each piece:

$$\mathcal{L}_{\text{balance}} = \sum_{k=1}^{K} \sum_{m=1}^{M} \left(S_{km} - \frac{\sum_{l=1}^{M} S_{kl}}{M} \right)^2. \quad (5)$$

This term also helps to avoid local minima where large cells poorly approximate the boundary (Fig. 7).

5 Geometric Dissection

Starting from the optimized Voronoi diagrams for shapes A and B, which roughly satisfy our requirements for dissection (Sec. 3), we refine the solution to generate an actual dissection by enforcing the isometry requirement while maintaining the topological structure

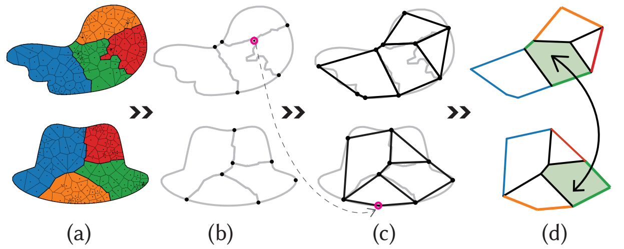


Fig. 4. Coarsening geometric dissection. We first coarsen the Voronoi diagram (a): contract boundary segments between corners where multiple cells meet (b), then perform common refinement of the graphs to make sure in each pair of corresponding faces vertices have 1-1 correspondence (c) (Pink vertex indicates a new inserted vertex). Finally, we optimize the locations of graph vertices, maintaining the bijectivity of this deformation (d).

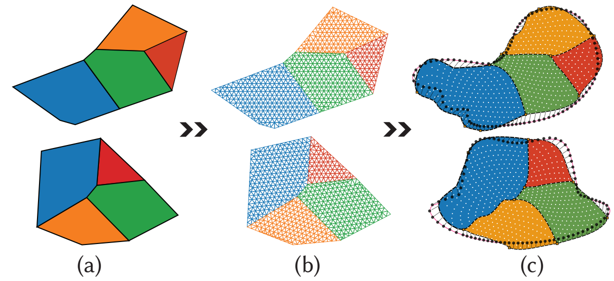


Fig. 5. We triangulate the coarse graphs from Sec. 5.1 (a,b), then bijectively deform the triangulations to better approximate the input shape (c). In (c), the input shapes are visualized in magenta; we show the correspondences between the arrangement boundary and the input shapes in black.

(adjacency of the pieces). Given the topology, optimizing all the piece shapes together remains difficult because it is a joint optimization of shape and rigid transformations. We develop a custom optimization strategy that proceeds in a coarse-to-fine manner by first leveraging the partitions to construct a feasible solution — a geometric dissection of simplified polygons (Sec. 5.1). We then refine this simplified dissection by bijectively deforming it to better match the input shapes, while keeping the found rigid transformations fixed (Sec. 5.2).

5.1 Coarse Geometric Dissection

The goal of this stage is to bijectively deform the partitions into a geometric dissection that coarsely approximates the input shapes. For the two partitions, we form an embedded graph where the vertices are the corners where multiple pieces meet, and edges connect corners to split adjacent pieces (Fig. 4a). This effectively coarsens each piece into a simple polygon. We then deform the polygons of corresponding pieces of the two partitions such that they become identical under rigid transformation, while preserving their connectivity.

Common refinement. Coarsening both partitionings independently may lead to corresponding polygons having a different number of vertices. To address this, we leverage the transformation of the pieces computed in Sec. 4. Given this transformation, we iterate over all polygons and insert vertices as necessary until all vertices of each polygon have a close correspondence in the corresponding polygon. Please refer to Alg. 2 in the Supplementary Material for details.

Bijective deformation of the coarse graphs. Let us denote $\bar{\mathcal{P}}^A$ as the set of vertices of the coarse polygon approximation for shape A, and similarly, $\bar{\mathcal{P}}^B$ is the corresponding set of points for shape B. Now that the polygons have one-to-one vertex correspondence, we are ready to formulate the optimization problem as

$$\underset{\bar{\mathcal{P}}^A,\bar{\mathcal{P}}^B,\mathbf{Q}_1,\ldots,\mathbf{Q}_K}{\arg\min} \quad \lambda_{\text{shape}}\mathcal{L}_{\text{shape}} + \lambda_{\text{int}}\mathcal{L}_{\text{int}} \qquad \text{s.t. } \mathbf{Q}_k(\bar{\mathbf{p}}^A_{km}) = \bar{\mathbf{p}}^B_{km}, \tag{6}$$

where $k = \{1,\ldots,K\}$, $\mathbf{Q}_k \in SE(2)$, and $\bar{\mathbf{p}}^A_{km} \in \bar{\mathcal{P}}^A$ and $\bar{\mathbf{p}}^B_{km} \in \bar{\mathcal{P}}^B$ are the m^{th} vertex position of the coarse approximated polygon of k^{th} piece for shapes A and B. Here $\mathcal{L}_{\text{shape}}$ aims to preserve the 2D length of the edges connecting the vertices of the polygons with all the possible combinations

$$\mathcal{L}_{\text{shape}} = \sum_{k=1}^{K} \sum_{\{e_1,e_2\}\in\mathcal{E}_k} \left(\|\bar{\mathbf{p}}^A_{ke_1} - \bar{\mathbf{p}}^A_{ke_2}\|_2 - \|\bar{\mathbf{p}}^B_{ke_1} - \bar{\mathbf{p}}^B_{ke_2}\|_2 \right)^2, \tag{7}$$

where the $\mathcal{E}_k$ is the set of edges inside the polygon k. The second term $\mathcal{L}_{\text{int}} = -(d-\hat{d})^2 \ln(d/\hat{d})$ is a barrier term that prevents intersection between edges; d is the minimum distance between two edges; $\hat{d}$ is the threshold distance — the barrier is only active when $d < \hat{d}$, and equals zero otherwise. This is necessary to keep the deformation bijective. The equality constraint enforces the isometry.

We solve this optimization via the penalty method [Nocedal and Wright 2006]; in our implementation, we set $\lambda_{\text{int}} = 300$; λ_{shape} and $\hat{d}$ are given in Suppl. 4. The penalty weight starts at 1.0 and doubles across four successive steps. At each step, our problem becomes unconstrained nonlinear least squares, so we solve it via 25 iterations of Levenberg–Marquardt with damping parameter $\lambda = 10^{-3}$, updated each iteration as $\lambda \leftarrow \max(\lambda/2,\ 10^{-6})$ on acceptance (if Armijo rule is satisfied), and $\lambda \leftarrow \min(10\lambda,\ 10^{12})$ on rejection. To enforce the hard constraint in Eq. (6), in the end, we fix the rigid transformations $\mathbf{Q}_k$ and orthogonally project the found solution onto the now linear equality constraint set (Fig. 4c). Note that fixing $\mathbf{Q}_k$ might lead to incompatible constraints (Eq. (6)) or self-intersections in the graph. We discard such cases and continue with the next initialization of the Voronoi sites (Sec. 4).

5.2 Bijective Refinement

Now that the rigid transformations between pieces are known, we refine the shapes of those pieces via another bijective deformation, bringing the assembled shapes closer to the input ones. We triangulate both graphs, maintaining consistent triangulation for the corresponding pieces (Fig. 5), then bijectively deform the two triangulations to better approximate the input shapes, while preserving the isometry and the seamlessness constraints. Let us denote $\tilde{\mathcal{P}}^A$ as the set of vertices of the triangulated polygon for shape A, and similarly, $\tilde{\mathcal{P}}^B$ is the corresponding set of points for shape B. We minimize

$$\underset{\tilde{\mathcal{P}}^A,\tilde{\mathcal{P}}^B}{\arg\min}\ \mathcal{L}_{\text{boundaryDist}} + \lambda_{\text{lap}}\mathcal{L}_{\text{lap}} + \lambda_{\text{SD}}\mathcal{L}_{\text{symmDirichlet}}$$
$$\text{s.t. } \mathbf{Q}_k(\tilde{\mathbf{p}}^A_{km}) = \tilde{\mathbf{p}}^B_{km}, \tag{8}$$

where $\tilde{\mathbf{p}}^A_{km} \in \tilde{\mathcal{P}}^A$ and $\tilde{\mathbf{p}}^B_{km} \in \tilde{\mathcal{P}}^B$ are the m^{th} vertex position of the triangulated polygon of the k^{th} piece for shapes A and B.

The equality constraints are linear, as the rigid transformations are now fixed. We remove those constraints by reparameterization, i.e., effectively substituting $\tilde{\mathbf{p}}^B_{km}$ into the objective function. We then follow Smith and Schaefer [2015], minimizing (8) via gradient descent with the line search step limited by the bijectivity constraint.

The boundary distance $\mathcal{L}_{\text{boundaryDist}}$ is the L_2 distance between the boundary of each arrangement and the corresponding input shape, both parameterized by arc length.

The second term in Eq. (8) leverages this arclength-based pointwise correspondence to encourage the Laplacian coordinates [Sorkine et al. 2004] of the boundaries of both arrangements to be close to the ones of the input shapes. Let $\delta_j^{\{A,B\}} \in \mathbb{R}^2$ be Laplacian coordinates of the j^{th} vertex of the boundary of arrangement A or B, computed by applying a 1D Laplacian to the vertex coordinates, and $\hat{\delta}_j^{\{A,B\}}$ be the Laplacian coordinates of the corresponding vertex on the input shape. Then

$$\mathcal{L}_{\text{lap}} = \sum_j \|\delta_j^A - \hat{\delta}_j^A\|_2^2 + \sum_j \|\delta_j^B - \hat{\delta}_j^B\|_2^2 \tag{9}$$

The last term, symmetric Dirichlet energy, regularizes the deformation by measuring triangle distortion. Per triangle t with a Jacobian F it is defined as:

$$\mathcal{L}_{\text{SD}}(t) \ = \ \|\mathbf{F}\|_F^2\left(1 + \frac{1}{\det(\mathbf{F})^2}\right). \tag{10}$$

The total distortion energy sums over all triangles weighted by their rest area S_t, normalized by the total mesh area:

$$\mathcal{L}_{\text{symmDirichlet}} \ = \ \frac{1}{S_{\text{total}}}\sum_t S_t\mathcal{L}_{\text{SD}}(t). \tag{11}$$

We perform 1000 gradient descent steps; we use $\lambda_{\text{lap}} = 2$, $\lambda_{\text{SD}} = 0.2$ for all experiments. The result of this optimization is our final result (Fig. 3d, Fig. 5c).

6 Results and Validation

We have automatically generated a number of geometric dissections (e.g., Figs. 1, 3, 13). We used the same parameter values for all the inputs in the paper. In the following, we further validate our method by comparing it with a previous approach (Sec. 6.1), via quantitative evaluations, and ablation studies (Sec. 6.2). Finally, we demonstrate our method in different application scenarios, including reuse of an existing shape and geometric dissections of multiple shapes.

6.1 Comparisons with Previous Work

In Figs. 13,14 and Table 1, we compare our method with the method of Duncan et al. [2017]. To ensure a fair runtime comparison, we set the time limit of our method to 1h, 4h, or 6h for the dissections into 4, 5, and 6 pieces, respectively; we always pick the result with the smallest average L_2 distance for the two shapes. We use an Intel Xeon Max 9480 CPU with 56 cores.

Qualitatively, compared with the automatic results of Duncan et al. [2017] (Fig. 13), while both methods visually reproduce the input shapes, ours generates compact and well-proportioned pieces due to the regularization term ($\mathcal{L}_{\text{reg}}$ in Sec. 4). In contrast, their method often produces elongated components with extremely narrow sections that might be brittle or harder to fabricate.

Quantitatively, our method has lower geometric error on nearly all examples (Table 1). We normalize all input shapes to the area of 10,000 px^2. We uniformly sample 100 points on each shape; we report Chamfer and Hausdorff distances to the input shapes. Our method has a slightly higher mean (Chamfer) distance on the Bunny–Egg example, where their method has a significantly higher maximum (Hausdorff) error (16.41 vs. 11.80). Each reconstructed dissection is then translated to best align with the input shape via ICP (keeping the rotations fixed).

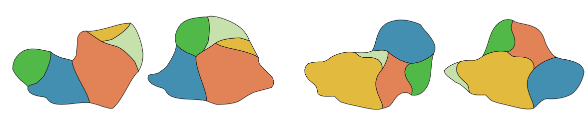
Even compared to the manually refined solutions of Duncan et al. [2017] (Fig. 14), our automatic approach consistently generates more compact and fabrication-friendly pieces while maintaining competitive or lower reconstruction error (Table 1).

6.2 Ablation Studies

We evaluate the contribution of two terms that control areas in the Voronoi-based partitioning (Sec. 4): the area balance term and the area term in Figs. 7 and 8. Without the balance term the partitioning might have large, irregular cells that poorly approximate the boundary geometry (Fig. 7). In Fig. 8, we average across 500 random initializations and report the mean and standard deviation of the average L_2 boundary distance to the input shapes, with each shape normalized to unit area. As the plot indicates, the area term speeds up the convergence and tends to produce local minima with lower loss values.

Number of Voronoi sites N. Increasing the number of Voronoi sites leads to a better approximation of the input shape (c.f. Fig. 9 with $N = 50$ vs. $N = 200$ Voronoi sites and Fig. 3 with $N = 125$ Voronoi sites), at the cost of higher computation time (75/80/94 min for 50/125/200 Voronoi sites, respectively).

Number of initializations for Voronoi-based partition. As discussed in Sec. 4, we run multiple Voronoi initializations in parallel to explore the solution space. Each successful initialization produces a valid dissection, so increasing the number of initializations trades additional computation for better approximation quality. As Figs. 6, 2, and 3(d) demonstrate, the best and second-best solutions are usually both visually plausible and closely approximate input shapes.



(a) Best solution (b) Second-best solution

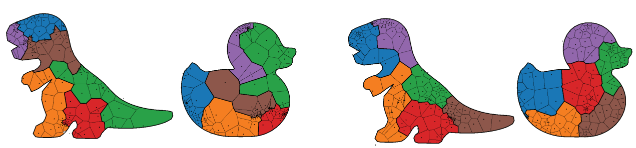
Fig. 6. Multiple 5-piece dissection solutions for Hat–Ghost case.

Shape Pair	First shape				Second shape				Averaged (first+second)		Time	
	Chamfer distance		Hausdorff distance		Chamfer distance		Hausdorff distance		Avg Chamfer	Avg Hausdorff	Duncan	Ours (sc)
	Duncan	Ours	Duncan	Ours	Duncan	Ours	Duncan	Ours	Duncan / Ours	Duncan / Ours		
Dog-Bone, 4 pieces	16.68	**2.65**	44.97	**7.99**	14.64	**2.43**	44.46	**9.78**	15.66 / **2.54**	44.72 / **8.89**	≈1h	1h (2100)
Dog-Bone, 5 pieces	6.20	**4.01**	14.11	**13.74**	8.83	**3.69**	17.25	**13.64**	7.52 / **3.85**	15.68 / **13.69**	N/A	4h (4688)
Trump-Map, 5 pieces	6.97	**2.57**	23.92	**9.52**	4.40	**3.76**	13.37	**12.83**	5.69 / **3.17**	18.65 / **11.18**	N/A	6h (5892)
Bunny-Egg, 6 pieces	5.34	**3.69**	16.41	**11.80**	**2.04**	2.83	**5.16**	7.68	3.69 / **3.26**	10.79 / **9.74**	≈9.7h	6h (4140)
Caterpillar-Butterfly, 6 pieces	5.85	**5.34**	17.54	**14.15**	6.70	**3.88**	19.11	**13.26**	6.28 / **4.61**	18.33 / **13.71**	N/A	6h (5643)
Cat-Bear, 6 pieces	5.22	**2.71**	17.79	**8.71**	4.10	**2.90**	10.33	**9.13**	4.66 / **2.81**	14.06 / **8.92**	N/A	6h (4285)
Serpent-Apple, 5 pieces	21.64	**9.05**	77.93	**46.99**	**3.73**	4.15	**9.83**	11.72	12.69 / **6.60**	43.88 / **29.36**	N/A	6h (5244)
Average (all)	–				–				8.03 / **3.83**	23.73 / **13.64**	–	–

Table 1. Comparison with Duncan et al. [2017] in terms of mean Chamfer distance, Hausdorff distance, and runtime. Our method achieves roughly half the error of Duncan et al. [2017] on average. The last column (sc) reports the number of successful configurations explored within the given runtime. Note that the best result can be identified at any point during the execution of our method.

Furthermore, the optimization is relatively insensitive to the choice of initialization. As shown in Fig. 8 (green), the Voronoi optimization exhibits stable convergence despite the non-convex objective. Across 500 random initializations, the best cost is 0.0130, while the mean and median are similar (0.0202 and 0.0196, respectively), with a standard deviation of 0.0038.

Number of pieces K. We next evaluate the influence of the number of pieces on success rate and computational cost. We run the optimization repeatedly for $K = 3, \ldots, 8$, collecting 3000 valid solutions per setting, with total run times of 85, 98, 133, 184, 308, and 417 minutes respectively. Results are shown in Figs. 15. Fig. 10(b) breaks down the runtime across pipeline stages: the Voronoi-based joint partitioning dominates the total cost. A GPU implementation may further speed up the Voronoi partitioning. Further analysis of the constraint chains is provided in Suppl. 5.



(a) Without area balance term (b) With area balance term

Fig. 7. Effect of the area balance term. Starting from the same initialization, the balance term helps regularize the cell area and avoid local minima in which large cells poorly approximate the boundary.

As discussed in Sec. 5.1, small inaccuracies in the estimated transformations during the final projection step of Eq. 6 may produce a self-intersecting coarse graph, causing the initialization to be discarded. Such failures become more frequent as the number of pieces increases. The successful initialization rate decreases from 95.0% for $K = 3$ to 49.8% for $K = 8$, with intermediate rates of 89.2%, 81.6%, 72.8%, and 62.0%. As a result, more initializations are required to obtain good solutions, leading to increased computation time.

6.3 Additional Applications

Shapes of different genus and multiple shapes. Due to the flexibility of the Voronoi-based joint partitioning, our method generalizes beyond the closed, one-to-one shape constraint of [Duncan et al. 2017]. As shown in Fig. 16(a), our method supports shapes with holes, demonstrated with a ghost-to-broccoli pair. It also handles one-to-multiple shape configurations, as demonstrated in Fig. 16(b) with a turtle-to-ghost and an arm example.

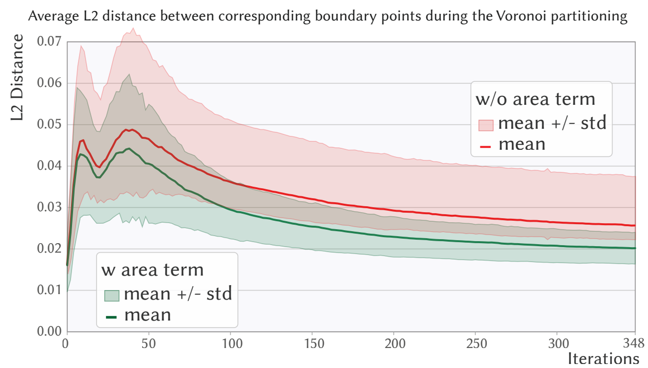


Fig. 8. **Ablation of the area term over 500 random initializations on hat–ghost 4-piece design.** Mean and ±1 standard deviation (shaded) of the average L_2 distance between corresponding boundary points—shapes normalized to unit area—as a function of optimization iterations. Without the area term (red), convergence is slower and yields a higher final error and larger variance. Including the area term (green) leads to faster convergence and a lower final error.

Shape reuse. Our method can also be used as a computational framework for modular fashion, which aims to create garments composed of interchangeable, reconfigurable components to support flexibility, customization, and sustainability [Del Valle et al. 2025; Li et al. 2018a; Lin et al. 2025]. As illustrated in Fig. 17, given two input garment panels (a), our approach generates a set of patches that can be reconfigured to assemble either a top or a pair of pants. Furthermore, adding hard constraints to keep vertices exactly on one of the input shapes, the pattern of the pants remains entirely unchanged during optimization, enabling complete panel reuse [Qi et al. 2025]. As shown in (c), the pants are fully reused into a top.

7 Limitations and Future Work

Our method only supports rotations and translations for the rigid transformations, but not reflections; including them in the search space could potentially improve shape approximation.

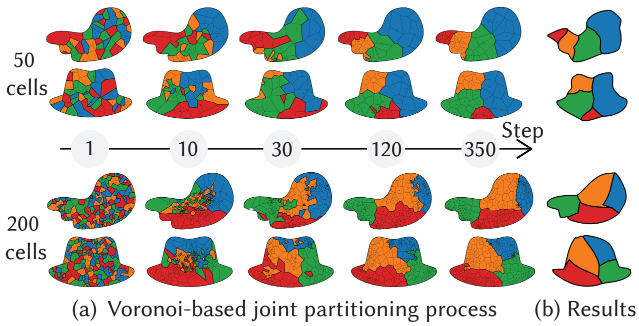


Fig. 9. (a) The first, 10th, 30th, 120th, and 350th iterations of Voronoi partitioning for 50 and 200 sites. We select the best result from 1,500 successful examples. (b) Final results. Using 200 sites yields a closer approximation to the input shape than 50 sites, at the cost of increased computation time.

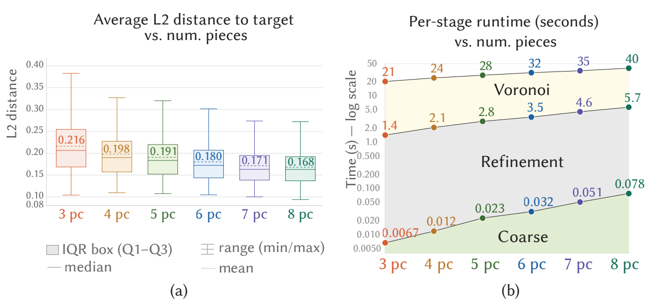


Fig. 10. Analysis of the effect of the number of pieces: (a) Average per-vertex L_2 distance to the target point vs number of pieces. Increasing the number of pieces improves the approximation accuracy; (b) Breakdown of the computational time across the different stages of the processing pipeline on a single thread.

Furthermore, while Li et al. [2018b] proposed a heuristic to test whether two shapes are good candidates for hinged dissection, it is unclear whether such a heuristic exists for our dissection problem; it could help identify promising shape pairs and warm-start our optimization. In practice, we observe that a *necessary condition* for a good approximate dissection is that the two shapes have similar distributions of boundary curvature (e.g., a spiral cannot be well matched to a circle). This reflects an inherent geometric limitation of the dissection problem itself rather than a limitation of our method. For the Ninja–Duck case in Fig. 11, the ninja's boundary contains many high-curvature regions distributed across the shape, whereas the duck's boundary is dominated by long, low-curvature segments with only a few localized high-curvature features (e.g., the beak and the tail). This mismatch in curvature distributions makes it challenging to reuse the same set of dissection pieces for both shapes.

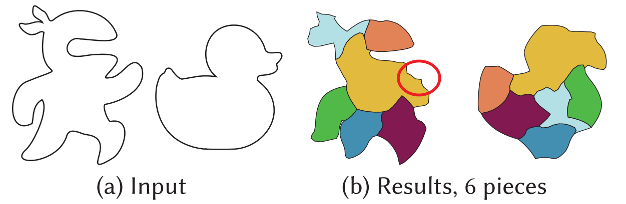


Fig. 11. Ninja–Duck, 6 pieces. The red circle highlights a region where the solution deviates noticeably from the input shape.

As future work, one could develop a more robust solver for Eq. 6 to improve scalability, as discussed in Sec. 6.2. Furthermore, as the number of pieces increases, some generated pieces become thin (red circles in Fig. 12(b)), making them difficult to fabricate. This issue could be addressed by improving the regularization terms or adaptively adjusting their weights to encourage more fabrication-friendly piece shapes.

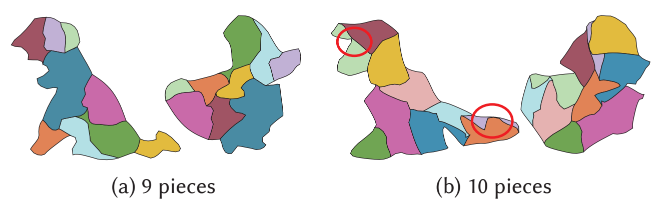


Fig. 12. Duck–Dinosaur dissection results using 9 and 10 pieces. We collect 500 valid solutions for each setting. The 9-piece case requires 98 minutes with a success rate of 32.9%, while the 10-piece case requires 162 minutes with a success rate of 17.2%. Red circles in (b) highlight the thin parts.

Our current optimization focuses on geometric approximation quality. Future work could include optimizing desirable properties for engaging puzzles, such as visually distinct, aesthetically pleasing, and interestingly shaped pieces, or controlling the target cognitive difficulty level of a puzzle.

Finally, extending our framework to 3D could unlock applications such as surface reuse [Baas et al. 2025] and modular surface design [Chen et al. 2023; Eigensatz et al. 2010]. A generalization to volumetric shapes appears relatively direct, whereas extending the method to surface meshes is substantially more challenging, as corresponding pieces must preserve both boundary geometry and surface curvature. We leave this problem for future work.

Acknowledgments

We thank Yuki Koyama and Koya Narumi for fabricating our dissections. We thank the reviews for their detailed feedback. This work was partially supported by JST ASPIRE JPMJAP2401, the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant No.: RGPIN-2024-04968, the NSERC - Fonds de recherche du Québec - Nature et technologies (FRQNT) NOVA Grant No. 314090, FRQNT team grant No. 361570, and a gift from Adobe.

References

Timothy G Abbott, Zachary Abel, David Charlton, Erik D Demaine, Martin L Demaine, and Scott D Kominers. 2012. Hinged dissections exist. *Discrete & Computational Geometry* 47, 1 (2012), 150–186.

Berend Baas, David Bommes, and Adrien Bousseau. 2025. Shape Approximation by Surface Reuse. *Computer Graphics Forum (Proc. Eurographics Symposium on Geometry Processing)* 44, 5 (2025), e70204. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.70204 doi:10.1111/cgf.70204

Farkas Bolyai. 1832. *Tentamen Juventutem Studiosam in Elementa Matheseos Purae Elementaris ac Sublimioris Methodo Intuitiva Evidentiaque Huic Propria Introducendi.* Typis Collegii Reformatorum, Marosvásárhely.

Jeffrey Bosboom, Erik D. Demaine, Martin L. Demaine, Jayson Lynch, Pasin Manurangsi, Mikhail Rudoy, and Anak Yodpinyanee. 2015. k-piece dissection is NP-hard. In *Abstracts from the 18th Japan Conference on Discrete and Computational Geometry and Graphs (JCDCGG 2015).*

Jan Brütting, Joseph Desruelle, Gennaro Senatore, and Corentin Fivet. 2019. Design of Truss Structures Through Reuse. *Structures* 18 (April 2019). doi:10.1016/j.istruc.2018.11.006

Oleksiy Busaryev, Tamal K. Dey, Huamin Wang, and Zhong Ren. 2012. Animating bubble interactions in a liquid foam. *ACM Trans. Graph.* 31, 4, Article 63 (July 2012), 8 pages. doi:10.1145/2185520.2185559

Rulin Chen, Pengyun Qiu, Peng Song, Bailin Deng, Ziqi Wang, and Ying He. 2023. Masonry Shell Structures with Discrete Equivalence Classes. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 42, 4 (2023). doi:10.1145/3592095

Sofia Colabella, Bernardino D'amico, Endrit Hoxha, and Corentin Fivet. 2017. Structural design with reclaimed materials: an elastic gridshell out of skis. In *Proceedings of IASS Annual Symposia*, Vol. 2017. International Association for Shell and Spatial Structures (IASS), 1–10.

Jurek Czyzowicz, Evangelos Kranakis, and Jorge Urrutia. 2007. Rectilinear Glass-Cut Dissections of Rectangles to Squares. https://api.semanticscholar.org/CorpusID:6408827

Fernando de Goes, Corentin Wallez, Jin Huang, Dmitry Pavlov, and Mathieu Desbrun. 2015. Power particles: an incompressible fluid solver based on power diagrams. *ACM Trans. Graph.* 34, 4, Article 50 (July 2015), 11 pages. doi:10.1145/2766901

Ashley Del Valle, Jennifer Jacobs, and Emilie Yu. 2025. texTile: Making and Re-making Crochet Granny Square Garments Through Computational Design and 3D-printed Connectors. In *Proceedings of the 2025 ACM Designing Interactive Systems Conference (DIS '25)*. Association for Computing Machinery, New York, NY, USA, 2445–2464. doi:10.1145/3715336.3735819

Erik D. Demaine, Tonan Kamata, and Ryuhei Uehara. 2026. Dudeney's Dissection Is Optimal. In *17th Innovations in Theoretical Computer Science Conference (ITCS 2026) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 362)*, Shubhangi Saraf (Ed.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany. doi:10.4230/LIPIcs.ITCS.2026.47

Qiang Du, Vance Faber, and Max Gunzburger. 1999. Centroidal Voronoi tessellations: Applications and algorithms. *SIAM review* 41, 4 (1999), 637–676.

Noah Duncan, Lap-Fai Yu, Sai-Kit Yeung, and Demetri Terzopoulos. 2017. Approximate dissections. *ACM Transactions on Graphics (SIGGRAPH Asia)* 36, 6, Article 182 (Nov. 2017), 13 pages. doi:10.1145/3130800.3130831

Michael Eigensatz, Martin Kilian, Alexander Schiftner, Niloy J. Mitra, Helmut Pottmann, and Mark Pauly. 2010. Paneling Architectural Freeform Surfaces. *ACM Transactions on Graphics* 29, 4, Article 45 (2010), 10 pages.

Andrea Favilli, Francesco Laccone, Paolo Cignoni, Luigi Malomo, and Daniela Giorgi. 2025. Optimizing Free-Form Grid Shells with Reclaimed Elements under Inventory Constraints. *Computer Graphics Forum (Proc. Eurographics)* (2025). doi:10.1111/cgf.70047

Fan Feng, Shiying Xiong, Ziyue Liu, Zangyueyang Xian, Yuqing Zhou, Hiroki Kobayashi, Atsushi Kawamoto, Tsuyoshi Nomura, and Bo Zhu. 2023. Cellular topology optimization on differentiable Voronoi diagrams. *Internat. J. Numer. Methods Engrg.* 124, 1 (2023), 282–304. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.7121 doi:10.1002/nme.7121

Greg N Frederickson. 2002. *Hinged dissections: swinging and twisting.* Cambridge University Press.

Chi-Wing Fu, Chi-Fu Lai, Ying He, and Daniel Cohen-Or. 2010. K-Set tilable surfaces. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 29, 4 (2010). doi:10.1145/1778765.1778781

Paul Gerwien. 1833. Zerschneidung jeder beliebigen Anzahl von gleichen geradlinigen Figuren in dieselben Stücke. *Journal für die reine und angewandte Mathematik* 10 (1833), 228–234.

Shrisudhan Govindarajan, Daniel Rebain, Dor Verbin, Kwang Moo Yi, Anish Prabhu, and Andrea Tagliasacchi. 2026. Power Foam: Unifying Real-Time Differentiable Ray Tracing and Rasterization. arXiv:2604.24994 [cs.GR] https://arxiv.org/abs/2604.24994

Laki Iinbor, Zhiyang Dou, and Wojciech Matusik. 2026. Soft Anisotropic Diagrams for Differentiable Image Representation. *ACM Transactions on Graphics (Proc. SIGGRAPH)* (2026).

Jelle Joustra, Bas Flipsen, and Ruud Balkenende. 2021. Structural reuse of wind turbine blades through segmentation. *Composites Part C: Open Access* 5 (2021), 100137. doi:10.1016/j.jcomc.2021.100137

Naoki Kita. 2023. Dissection puzzles composed of multicolor polyominoes. In *Computer Graphics Forum*, Vol. 42. Wiley Online Library, e14968.

Robert Kovacs, Anna Seufert, Ludwig Wall, Hsiang-Ting Chen, Florian Meinel, Willi Müller, Sijing You, Maximilian Brehm, Jonathan Striebel, Yannis Kommana, Alexander Popiak, Thomas Bläsius, and Patrick Baudisch. 2017. TrussFab: Fabricating Sturdy Large-Scale Structures on Desktop 3D Printers. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 2606–2616. doi:10.1145/3025453.3026016

Bruno Lévy and Yang Liu. 2010. Lp Centroidal Voronoi Tessellation and its applications. In *ACM SIGGRAPH 2010 Papers* (Los Angeles, California) *(SIGGRAPH '10)*. Association for Computing Machinery, New York, NY, USA, Article 119, 11 pages. doi:10.1145/1833349.1778856

Meng-Mi Li, Ying Yan Chen, and Yedong Wang. 2018a. Modular Design in Fashion Industry. https://api.semanticscholar.org/CorpusID:62790998

Shuhua Li, Ali Mahdavi-Amiri, Ruizhen Hu, Han Liu, Changqing Zou, Oliver Van Kaick, Xiuping Liu, Hui Huang, and Hao Zhang. 2018b. Construction and fabrication of reversible shape transforms. *ACM Trans. Graph.* 37, 6, Article 190 (Dec. 2018), 14 pages. doi:10.1145/3272127.3275061

Rebecca Lin, Michal Lukáč, and Mackenzie Leake. 2025. Refashion: Reconfigurable Garments via Modular Design. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. Busan, Republic of Korea. doi:10.1145/3746059.3747632

Yang Liu, Wenping Wang, Bruno Lévy, Feng Sun, Dong-Ming Yan, Lin Lu, and Chenglei Yang. 2009. On centroidal voronoi tessellation—energy smoothness and fast computation. *ACM Trans. Graph.* 28, 4, Article 101 (Sept. 2009), 17 pages. doi:10.1145/1559755.1559758

Lin Lu, Andrei Sharf, Haisen Zhao, Yuan Wei, Qingnan Fan, Xuelin Chen, Yann Savoye, Changhe Tu, Daniel Cohen-Or, and Baoquan Chen. 2014. Build-to-last: strength to weight 3D printed objects. *ACM Trans. Graph.* 33, 4, Article 97 (July 2014), 10 pages. doi:10.1145/2601097.2601168

Eli Maor. 2007. *The Pythagorean Theorem: A 4,000-Year History.* Princeton University Press.

Yuxuan Mei, Benjamin Jones, Dan Cascaval, Jennifer Mankoff, Etienne Vouga, and Adriana Schulz. 2024. FabHacks: Transform Everyday Objects into Home Hacks Leveraging a Solver-aided DSL. In *Proceedings of the 9th ACM Symposium on Computational Fabrication* (Aarhus, Denmark) *(SCF '24)*. Association for Computing Machinery, New York, NY, USA, Article 4, 16 pages. doi:10.1145/3639473.3665788

Andriy Myronenko and Xubo Song. 2010. Point Set Registration: Coherent Point Drift. *IEEE Trans. Pattern Anal. Mach. Intell.* 32, 12 (Dec. 2010), 2262–2275. doi:10.1109/TPAMI.2010.46

Jorge Nocedal and Stephen J. Wright. 2006. *Numerical optimization* (2. ed. ed.). Springer, New York, NY.

Logan Numerow, Yue Li, Stelian Coros, and Bernhard Thomaszewski. 2024. Differentiable Voronoi Diagrams for Simulation of Cell-Based Mechanical Systems. *ACM Trans. Graph.* 43, 4, Article 69 (July 2024), 11 pages. doi:10.1145/3658152

A. Robert Pargeter. 2004. Dissections: plane and fancy. *The Mathematical Gazette* 88, 512 (2004), 371–373. arXiv:https://doi.org/10.1017/S002555720017559X doi:10.1017/S002555720017559X

Jesse Pupping, Marzia Riso, Mariana Popescu, Adrien Bousseau, and Jelle Joustra. 2025. From blades to tracks: a case study in structural reuse of curved surfaces for circular design. In *ACM Symposium on Computational Fabrication (SCF)*. ACM. http://www-sop.inria.fr/reves/Basilic/2025/PRPBJ25

Anran Qi, Nico Pietroni, Maria Korosteleva, Olga Sorkine-Hornung, and Adrien Bousseau. 2025. Rags2Riches: Computational Garment Reuse. *SIGGRAPH 2025, Technical Papers* (2025). doi:10.1145/3721238.3730703

Mayank Singh and Scott Schaefer. 2010. Triangle surfaces with discrete equivalence classes. *ACM Transactions on Graphics* 29, 4, Article 46 (July 2010), 7 pages. doi:10.1145/1778765.1778783

Jason Smith and Scott Schaefer. 2015. Bijective parameterization with free boundaries. *ACM Trans. Graph.* 34, 4 (2015), 70–1.

Peng Song, Chi-Wing Fu, Yueming Jin, Hongfei Xu, Ligang Liu, Pheng-Ann Heng, and Daniel Cohen-Or. 2017. Reconfigurable interlocking furniture. *ACM Transactions on Graphics (SIGGRAPH Asia)* 36, 6, Article 174 (Nov. 2017), 14 pages. doi:10.1145/3130800.3130803

Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. 2004. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. 175–184.

Keke Tang, Peng Song, Xiaofei Wang, Bailin Deng, Chi-Wing Fu, and Ligang Liu. 2019. Computational Design of Steady 3D Dissection Puzzles. *Computer Graphics Forum*

(2019). doi:10.1111/cgf.13638

Davide Tonelli, Nico Pietroni, Enrico Puppo, Maurizio Froli, Paolo Cignoni, Gennaro Amendola, and Roberto Scopigno. 2016. Stability of statics aware voronoi grid-shells. *Engineering Structures* 116 (2016), 70–82.

Ziqi Wang, Peng Song, and Mark Pauly. 2021. State of the Art on Computational Design of Assemblies with Rigid Parts. *Computer Graphics Forum (Proc. of Eurographics)* (2021).

Xuanyu Wu, Kenji Tojo, and Nobuyuki Umetani. 2024. Free-form Floor Plan Design using Differentiable Voronoi Diagram. In *Pacific Graphics Conference Papers and Posters*. The Eurographics Association. doi:10.2312/pg.20241296

Yahan Zhou and Rui Wang. 2012. An Algorithm for Creating Geometric Dissection Puzzles. In *Proceedings of Bridges 2012*. Tessellations Publishing, 49–56. http://archive.bridgesmathart.org/2012/bridges2012-49.html

Henrik Zimmer, Florent Lafarge, Pierre Alliez, and Leif Kobbelt. 2014. Zometool shape approximation. *Graphical Models* 76, 5 (2014), 390–401. doi:10.1016/j.gmod.2014.03.009 Geometric Modeling and Processing 2014.

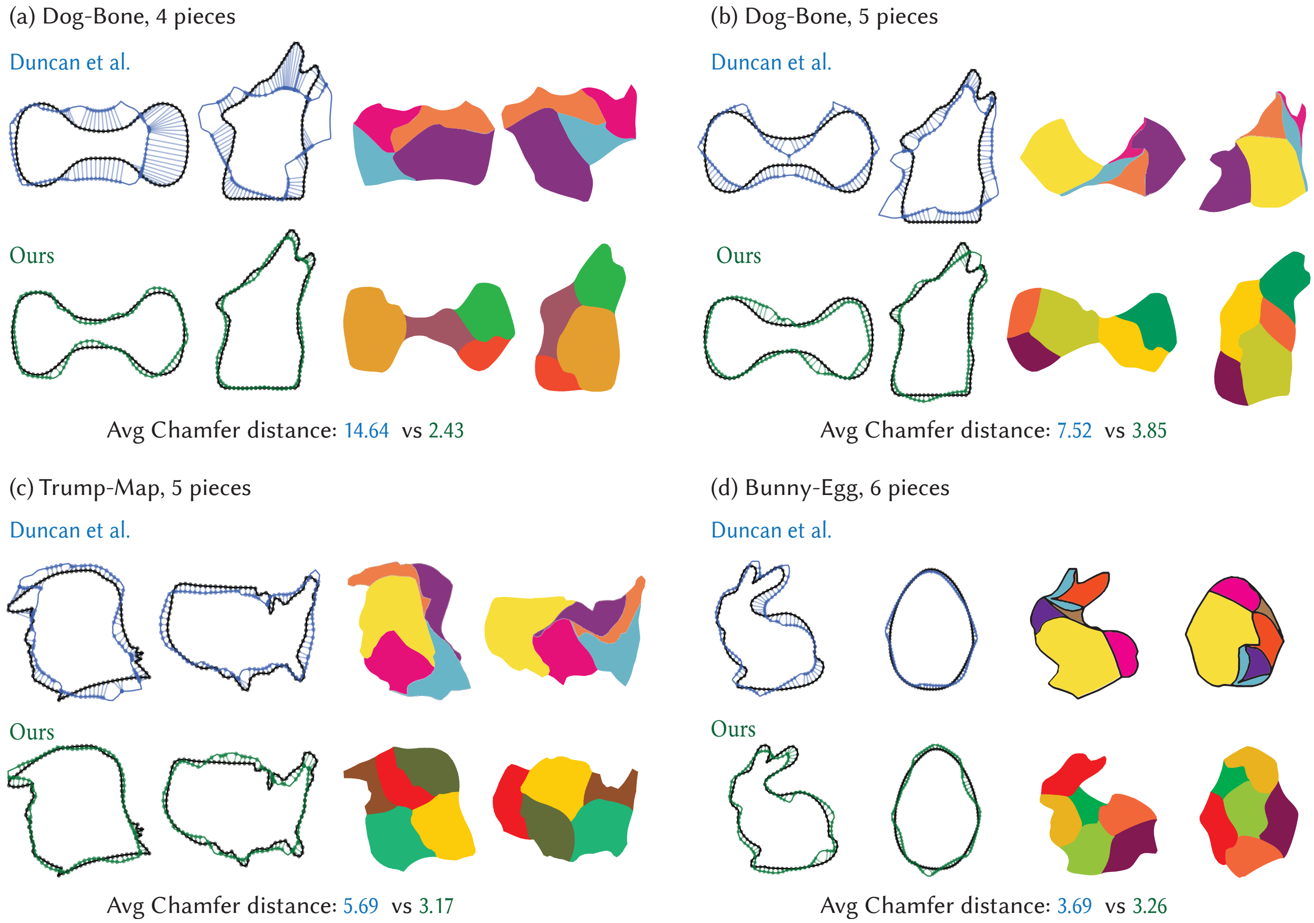


Fig. 13. Comparison with the automatic results of Duncan et al. [2017]. The black shapes are the original inputs; to the right of each case is the registered point correspondence. Our approach achieves lower geometric error on nearly all examples, except for the egg, where [Duncan et al. 2017] attains a smaller error at the cost of large distortion on the bunny.

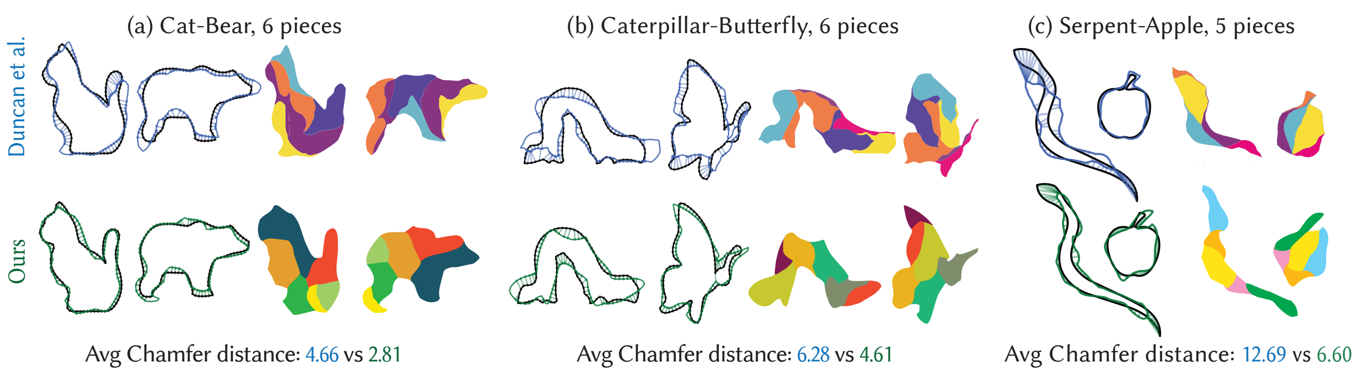


Fig. 14. Comparison against the user-guided version of Duncan et al. [2017], which relies on manual annotations, whereas our method remains fully automatic throughout.

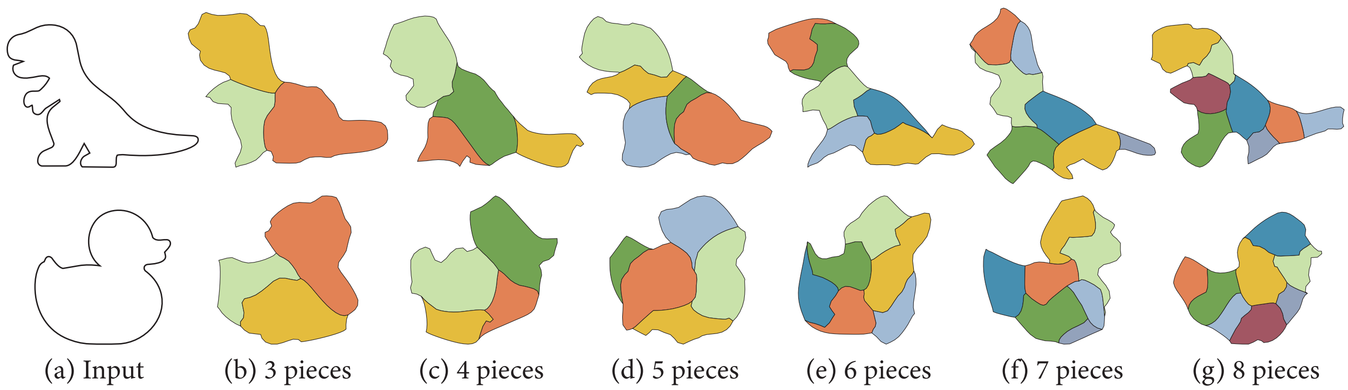


Fig. 15. Duck-Dinosaur dissection results using 3-8 pieces.

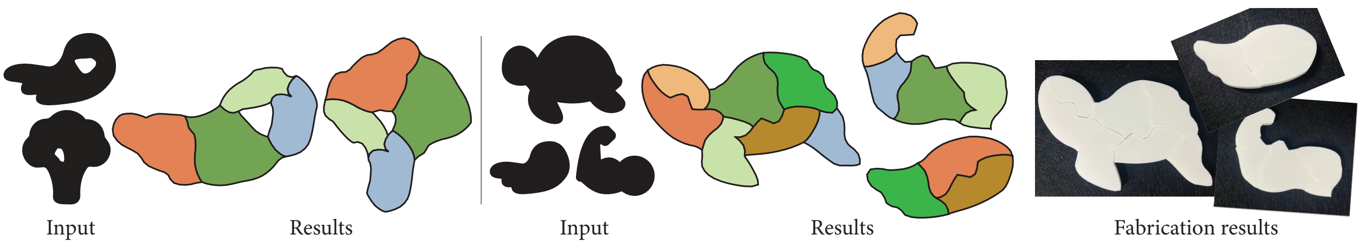


Fig. 16. Left: Shapes containing holes: Ghost–Broccoli; Right: Shapes composed of multiple disconnected components: Turtle–Ghost.

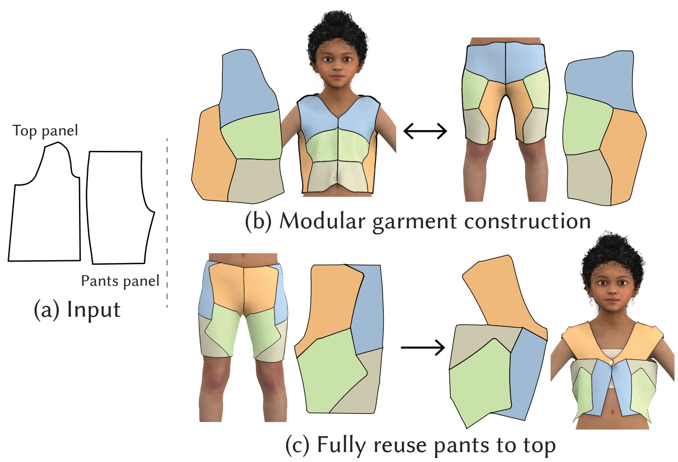


Fig. 17. Given two input garment panels (a), (b) our method generates patches that can be reconfigured into either a top or a pair of pants, enabling modular garment design. (c) By keeping the pant panel unchanged during optimization, it can be directly reused to a top with no material waste. Note in both cases the top and pants can not co-exist.