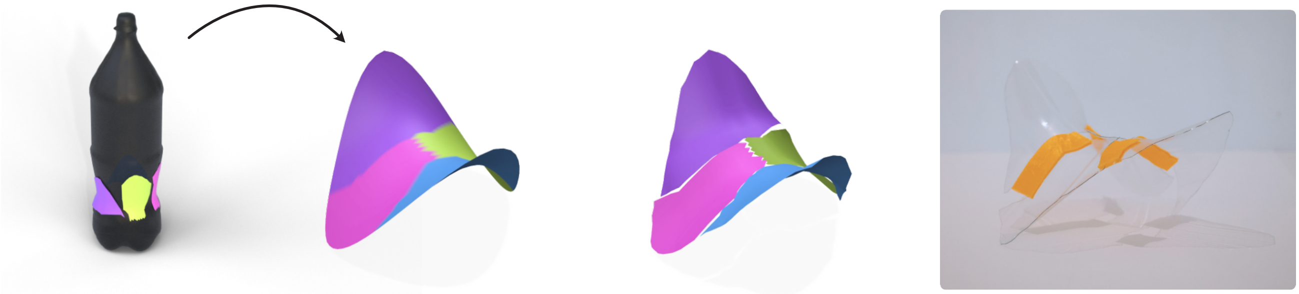
Eurographics Symposium on Geometry Processing 2025
M. Attene and S. Sellán
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 44 (2025), Number 5

Shape Approximation by Surface Reuse

Berend Baas[1], David Bommes[2], Adrien Bousseau[1]

[1] Inria Université Côte d'Azur, France
[2] University of Bern, Switzerland



(a) Source (b) Target (c) Reconstruction (d) Fabricated Result

Figure 1: *Our method decomposes source and target shapes (a,b) to approximate the target with panels taken from the source (c). In this example, we fabricate a model of a saddle using panels cut from a plastic bottle and assembled with 3D-printed connectors (d).*

Abstract

The manufacturing industry faces an urgent need to transition from the linear "make-take-use-dispose" production model towards more sustainable circular models that retain resources in the production chain. Motivated by this need, we introduce the new problem of approximating 3D surfaces by reusing panels from other surfaces. We present an algorithm that takes as input one or several existing shapes and relies on partial shape registration to identify a small set of simple panels that, once cut from the existing shapes and transformed rigidly, approximate a target shape within a user-defined distance threshold. As a proof of concept, we demonstrate our algorithm in the context of rapid prototyping, where we harvest curved panels from plastic bottles and assemble them with custom connectors to fabricate medium-size freeform structures.

Keywords: circular design, reuse, shape matching, shape approximation, fabrication, rapid prototyping

1. Introduction

The circular economy aims at reducing waste and resource consumption by reusing the materials of discarded objects to produce new objects [BdvZ14]. Recycling achieves this goal by retrieving raw materials that can then be reprocessed. However, many materials lose quality through recycling, or cannot be recycled at all. *Structural reuse* is an alternative strategy to recycling that consists in cutting discarded objects (which we call *sources*) into parts that are re-assembled to form new objects (which we call *targets*) [FB20, JFB21]. Recent work in architectural geometry has applied this strategy to create structures by reusing beams [BDSF19, VMLC24, FLC*25] and tree forks [AHM*20, MS16].

In this paper, we go beyond linear elements and propose the problem of reusing rigid panels of 3D surfaces to approximate other surfaces. Reusing curved surfaces raises specific challenges not present in the reuse of linear elements. On the one hand, the presence of curvature greatly limits the possibilities of finding good matches between source and target panels. On the other hand, working with surfaces greatly increases the number of elements to be considered. Even with very coarse discretizations of the surfaces, the number of panels of arbitrary shape grows rapidly, preventing exhaustive evaluation of all possible matches.

We propose an initial approach to this problem, building on well-established tools from geometry processing. In particular, we take inspiration from prior work on partial shape registration [MGP06, CZ08] to make our problem tractable by splitting surface reuse into two sub-problems: We first generate a set of local candidate correspondences between the source and target, using a suitable local shape descriptor that is invariant to rigid transformations. We then phrase our optimization as a graph-cut problem, where the objective is to find a covering of the target surface from these initial candidate correspondences. To avoid the trivial solution of partitioning the surface into many, tiny panels, we incorporate a term to promote using a small number of panels [DOIB12]. We also include a smoothness term to penalize intricate panels that would be difficult to cut and re-assemble.

Our work is partly motivated by the reuse of large curved objects, such as windturbine blades, boat hulls and aircraft fuselages that are made of fiber-reinforced polymers that are difficult to recycle [LZW*25,JFB21,LB17,LGJ*17]. But for ease of implementation, we make a proof of concept on the smaller-scale scenario of reusing plastic bottles for rapid prototyping. For this use case, we describe a semi-automated fabrication pipeline, where the segmentation from our algorithm is used to generate the cutting patterns for the bottle, as well as a sparse set of 3d-printed connectors, which together form an "assembly kit" for the target shape. This fabrication procedure allows the creation of lightweight physical prototypes of curved surfaces that span tens of centimeters, while requiring much less raw material than would be required to 3D print.

In summary, we make the following contributions:

- We introduce the new problem of shape approximation by reuse.
- We describe a practical algorithm to approximate a target shape with panels cut from one or several existing shapes.
- We describe a fabrication procedure to prototype freeform shapes by cutting and assembling panels from plastic bottles.

2. Related Work

We first discuss digital tools for circular design, followed by related work in geometry processing and computational fabrication, including shape matching and shape approximation.

Circular design. The societal need for sustainable production models has recently driven researchers to look at applying computational methods to problems of sustainable and/or circular design. When manufacturing products from raw materials, numerical optimization can help reduce material waste [KHLM16,ZMB*24] or rationalize shapes to be made of a small number of standardized elements [CQS*23,EKS*10,SS10,FLHCO10]. Although standardization contributes to circularity by facilitating the repair and reuse of elements across structures, we target the opposite problem of reusing non-standard parts.

More related to our research are methods to reuse stocks of existing elements reclaimed from decommissioned structures to create new structures with target shapes or performances. The cutting and assignment of reclaimed beams [BDSF19,VMLC24,FLC*25] or tree forks and branches [AHM*20, MS16, YLI19] has been achieved through mixed-integer program or Hungarian algorithm formulations [THL23,HADWM21]. While these methods allow for the creation of truss structures and grid shells by reusing linear elements, they do not readily extend to curved surfaces due to the combinatorial explosion of candidate elements to be matched. In a concurrent work, Qi et al. [QPK*25] describe a computational approach to garment reuse. In contrast to the rigid shapes we target, fabric can be flattened, which Qi et al. exploit to search for matches between garment panels in a 2D quantized space. Closer to our target application domain, Joustra et al. [JFB21] study how to segment windturbine blades into flat panels within timber standards, but they define segmentation patterns manually and do not reuse highly-curved parts. Windturbine blades are estimated to result in a cumulative waste of tens of million tons by 2050 [LB17], calling for innovative reuse strategies to avoid incinerating or landfilling such high-grade, curved materials. Other industries would also benefit from reuse, such as aeronautics that is adopting fiber-reinforced polymers to produce lightweight fuselage panels, but is anticipated to result in 0.5 million tons of waste by 2050 [LGJ*17].

Another of our target applications is rapid prototyping through reuse of waste materials, such as plastic bottles. A similar idea was explored by Kovacs et al. [KSW*17], who describes an interactive system to build truss structures using bottles as beams. However, they treat bottles as elementary elements that they assemble into larger structures, while we explore how to cut bottles into arbitrary panels to prototype freeform shapes. Similarly, Mei et al. [MJC*24] present a system to create fixtures from everyday objects, but focus on creating functional object assemblies rather than cutting objects for shape approximation. We also share the motivation of Wall et al. [WJVS21] who use scrap materials instead of infill material to reduce the consumption of raw plastic for 3D printing. Similarly, by reusing plastic bottles we can fabricate curved surfaces while only 3D-printing small connectors.

Shape Approximation. Our application relates to prior work on approximating shapes with surface panels having specific properties, often motivated by manufacturing constraints. Representative examples include shape approximation with flat [CSaLM13] or developable panels [IRHS20, BVHSH21, SGC18, SAJ20], polynomials [YAB*22], heightfields [HMA15], inflatable structures [PIC*21], stretched or pleated fabric [JSVB22, SRVSH24]. A key distinction of our work is that our design space is not dictated by an analytical formula (planarity, developability), nor by a simulation process (inflation, stretching), but by a prescribed stock of material. Nevertheless, our algorithm shares similarities with some of the methods listed above, especially those that employ a graphcut optimization to assign each point of the target surface to one of a set of candidate panels [CSaLM13, IRHS20, HMA15, YAB*22].

Our work also relates to shape collage [GSP*07], reassembly of fractured objects [LBB12, XCH*24], and other assembly-based modeling systems [FKS*04, SFCH12] that approximate or reconstruct a shape using a pre-defined set of parts. While we build on similar shape matching algorithms, we differ in that in our context, the source parts are not known in advance and need to be discovered along with their placement over the target.

Finally, our work is related to so-called *dissection puzzles*, where the goal is to create several shapes made of a shared set of pieces. Recent papers have approached the problem of designing flat or volumetric dissection puzzles using polygonal [DYYT17, LMAH*18] or voxelized approximations [TSW*19, ZW12]. Similar approaches have been used to design reconfigurable furniture consisting of parts that can be assembled or transformed into multiple target designs [SFJ*17]. However, these methods jointly design several shapes such that they are reconfigurable, while in our setting, the source shape already exists and greatly constrains the target shapes than can be created from it.

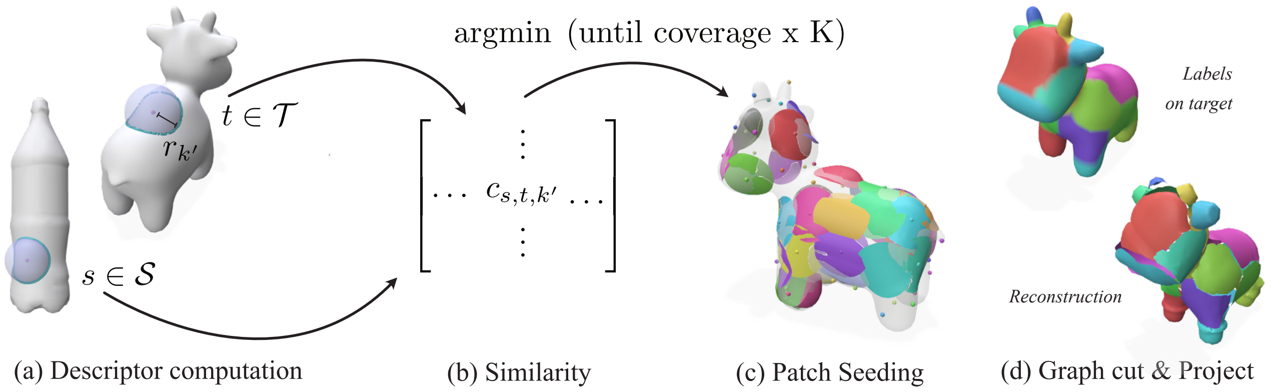


Figure 2: *Overview of our partitioning algorithm. We start by densely computing a set of multi-scale descriptors across the source and target shapes (a). From these descriptors we compute similarity between candidates at multiple scales (b), which we use to seed our graph-cut optimizer (c). The optimization then selects a subset of patches and grows these to cover the source within a user-specified threshold. We then reproject the solution to obtain the source parts (d).*

We refer the reader to the survey by Wang et al. [WSP21] for a broader discussion on assembly-based computational design.

Partial Shape Matching. The core of our approach consists in finding partial registrations between the source and target shapes, a topic that has received significant attention in geometry processing [vKZHCO11, DYDZ22]. In particular, our goal is to identify matching parts that are large and simple, which is a difficult optimization problem [BB08]. We achieve this goal by following prior work on partial symmetry detection [MGP06] and on registration of articulated shapes [CZ08], where matching parts are discovered by computing local shape descriptors over the two shapes, registering pairs of regions having similar descriptors, and aggregating neighboring regions that undergo the same transformation. Similarly to Chang and Zwicker [CZ08], we cast the aggregation step as a graphcut problem, which we augment with a *label cost* [DOIB12] to obtain a small number of parts.

3. Shape Approximation by Reuse

3.1. Problem Formulation

Our problem is to reconstruct a given target surface $\mathcal{T} \subset \mathbb{R}^3$ from rigid pieces cut from a (set of) source surfaces $\mathcal{S} \subset \mathbb{R}^3$. Specifically, we want to find a set of panels $\{\mathcal{S}_c \subset \mathcal{S}\}$ accompanied by rigid transformations $\{f_c \in \mathrm{SE}(3)\}$, such that the combination $\bigcup_c f_c(\mathcal{S}_c)$ yields our target surface $\mathcal{T}$. Since perfect reconstruction of the target is unlikely to exist, we aim at approximating the target according to a shape distance $\mathcal{L}_{\text{accuracy}}$ (we use the Chamfer distance). Furthermore, we want this approximation to be composed of a small number of panels to preserve the material integrity of the source and to reduce the cost of assembling the target. Finally, we also want simple panels with smooth boundaries to ease cutting. Our mathematical formulation combines these objectives:

$$\begin{aligned}\underset{\mathcal{S}_c \subset \mathcal{S}, f_c \in \mathrm{SE}(3)}{\arg\min} \quad & \mathcal{L}_{\text{accuracy}}\Big(\bigcup_c f_c(\mathcal{S}_c), \mathcal{T}\Big) \\ & + w_{\text{smoothness}} \mathcal{L}_{\text{smoothness}}(\{\mathcal{S}_c\}) \\ & + w_{\text{fragmentation}} \mathcal{L}_{\text{fragmentation}}(\{\mathcal{S}_c\}),\end{aligned} \tag{1}$$

where $\mathcal{L}_{\text{smoothness}}$ measures the simplicity of panel boundaries, and $\mathcal{L}_{\text{fragmentation}}$ counts the number of panels. Furthermore, we impose the following constraints on the solution:

- The panels should not overlap on the target to avoid material waste and to ensure connection at their boundaries.
- Any point of the reconstruction should be within a distance ϵ to the target to preserve the design intent. If this constraint cannot be satisfied, the corresponding part of the target will not be reconstructed.

Note that we allow parts of the source to be used multiple times, which means that multiple physical copies of the source may be needed to manufacture the target.

However, solving this optimization problem is a difficult task that combines discrete decisions (number of panels) and continuous variables (shape and transformation of each panel). Figure 2 illustrates the main steps of our method to perform this optimization in a scalable manner. We first compute local shape descriptors at each vertex of the source and target shapes. These descriptors allow us to drastically reduce the search space by identifying an over-complete set of promising matching source and target patches. Registering these patches provides us, for every vertex on the target mesh, a set of transformed candidate panels $\{f_c(\mathcal{S}_c)\}$ within the prescribed threshold distance ϵ. The last step of our algorithm consists in selecting one candidate panel per vertex (no overlap) to form the subset of panels that minimize Equation 1.

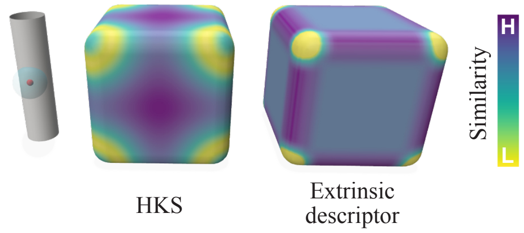


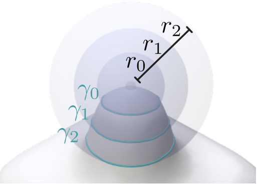
Figure 3: *Since intrinsic descriptors, like the Heat Kernel Signature (middle), are invariant under isometric deformation, they cannot distinguish flat surface patches (the faces of the cube) from developable ones (the rounded edges). Since we aim at matching rigid surface patches, we use a descriptor that is sensitive to extrinsic curvature (right, where only the rounded edges are identified as similar to the point on the source cylinder).*

3.2. Local shape descriptor

The literature on shape registration offers a vast choice of local shape descriptors [vKZHCO11, HPPLG11]. While many descriptors have been developed to be invariant to isometric deformations, our target application calls for a descriptor that is invariant to rigid transformations. Figure 3 illustrates the difference between intrinsic descriptors [SOG09] and the extrinsic descriptor we have adopted. Since intrinsic descriptors are invariant to isometry, they consider all developable surfaces to be similar, while we want to distinguish flat parts from cylindrical ones, for different degrees of curvature. Furthermore, we need a multiscale descriptor to match large patches when possible, falling back to smaller patches when unable to match larger ones. Finally, we need a descriptor that has a well-defined spatial extent, to ease subsequent patch registration.

Definition. The above requirements led us to adopt a descriptor based on *integral invariants* [PWHY09] that measure curve lengths or surface areas over small kernel domains. Specifically, we take inspiration from the work of Gehre et al. [GBK16] and Mortara et al. [MPS*04].

Similar to Gehre et al., we grow balls $\{B_p(r_k)\}$ centered at query point $p \in \{\mathcal{S}, \mathcal{T}\}$ on the surface of interest, for a set of increasing radii r_k spaced linearly between a user-set $r_{\min}$ and $r_{\max}$ radius. However, while Gehre et al. measure distances in geodesic space to be invariant to isometries, we follow Mortara et al. and measure distances in Euclidean rather than geodesic space to be invariant to rigid transformations.



In other words, we take the restricted patches of the surface intersecting these balls in the embedding space: $\Omega_{p,r_k} = B_p(r_k) \cap \mathcal{S}$, such that $p \in \Omega_{p,r_k}$ and Ω_{p,r_k} consists of a single connected component. Note that in the presence of thin structures, this procedure can yield patches with multiple boundary curves. We reject those candidates to only consider patches of disk topology during the registration in Section 3.3.

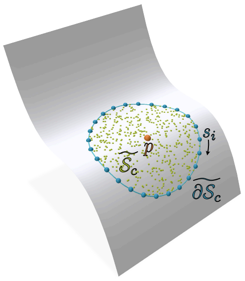
The resulting set of concentric patches captures information about the local curvature of the surface at multiple scales [PWHY09]. Specifically, the length of successive boundary curves grows linearly when the surface is flat, and varies non-linearly when the surface is curved. Following Gehre et al. [GBK16], we compute the arc-length $a(p, r_k) = |\partial\Omega_{p,r_k}|$ of each patch boundary, and define our descriptor at point p as the vector $\langle a(p, r_k)\rangle$. A useful property of this descriptor is that the patch boundary curves provide us with a fixed arc-length parametrization, which we later leverage for patch registration (Section 3.3).

Discretization. We extract the boundary of each patch Ω_{p,r_k} over the triangle mesh by tracing the isocurve of value r_k for the euclidean distance to p. For accuracy, we determine the location of the isocurve along triangle edges through line-sphere intersection, and compute the analytical length of each circular arc forming the isocurve segments crossing the triangles.

Matching. We measure the agreement between a pair of source and target points (s, t) by computing the cost function as the L_2 distance between their vectors of boundary lengths. In practice, we compute the distance up to each scale k' for $r_{\min} < r_{k'} < r_{\max}$ to obtain a cost for triples $(s, t, k') \in \mathcal{S} \times \mathcal{T} \times \mathbb{N}$:

$$d_{\text{Desc}}(s, t, k') = \sum_{k=1}^{k'} \|a(s, r_k) - a(t, r_k)\|^2. \tag{2}$$

3.3. Patch registration


p
s_i
$\widetilde{\mathcal{S}_c}$
$\widetilde{\partial\mathcal{S}_c}$

Since the descriptor introduced above estimates patch agreement based on boundary similarity (integrated across scales), we utilize this property to obtain a fast registration of a patch pair $(\mathcal{S}_c, \mathcal{T}_c)$. Specifically, we perform a boundary-first alignment, which we validate against the surface geometry to measure reconstruction accuracy. We start by sampling the boundary of the corresponding patches at regular arc-length subdivisions, obtaining $\widetilde{\partial\mathcal{S}_c} := (s_1, \dots, s_n) \subset \partial\mathcal{S}_c$ and $\widetilde{\partial\mathcal{T}_c} := (t_1, \dots, t_n) \subset \partial\mathcal{T}_c$. Additionally, we sample points from the surfaces of the patches $\widetilde{\mathcal{S}_c} \subset \mathcal{S}_c$, $\widetilde{\mathcal{T}_c} \subset \mathcal{T}_c$. To perform alignment, we consider all cyclic permutations σ_i of the sampled boundary $\widetilde{\partial\mathcal{S}_c}$, and perform Procrustes alignment on the pointwise (boundary) correspondences $\{s_{\sigma_i(j)}, t_j\}_{j=1\dots n}$ to obtain the transformation $f_i \in \mathrm{SE}(3)$ that best aligns the boundaries under cyclic permutation σ_i. For each obtained f_i we then validate the alignment against the sampled interior point clouds $\widetilde{\mathcal{S}_c}$ and $\widetilde{\mathcal{T}_c}$ using the Chamfer distance, and pick the f_i that yields the best alignment:

$$\arg\min_{f_i} \mathcal{L}_{\text{accuracy}}(f_i(\widetilde{\mathcal{S}_c}), \widetilde{\mathcal{T}_c}). \tag{3}$$

Note that in the presence of symmetries, several registrations might attain the minimum, we select one of them arbitrarily. When dealing with materials whose properties are independent of surface orientation, such as the plastic bottles we used in our experiments, we also traverse the source boundary in reverse permutation order σ_i^{-1} to consider alignment of the flipped patch.

Compared to ICP, leveraging the arc-length parameterization of our descriptor boundaries has the advantage of not requiring an initial guess, nor iterations to update point correspondences. Due to the independence of all permutations, patch registration is also easy to parallelize on the GPU. In practice, we use $n = 200$ boundary samples and $m = 5000$ interior samples in our experiments.

3.4. Automatic partitioning

The ingredients above allow us to find, for any vertex of the target, the best matching vertices on the source at multiple scales, and to quickly register the corresponding surface patches centered on those vertices. Equipped with these ingredients, our algorithm consists of two phases. In the seeding phase, we create a large set of candidate local registrations of the source to cover the target. In the partitioning phase, we select one candidate registration per target vertex to form large panels.

Seeding. Since we aim at reusing few, large panels, we create candidates by searching for good matches between source and target at large scale. However, large-scale matches might not suffice to achieve full coverage of the target. We thus also search for matches at smaller scale, even though the resulting panels might not extend to large portions of the shape, as visualized in Figure 4. At each scale k', we employ a greedy search that starts with the list of target vertices, selects the best matching pair of descriptors according to Equation 2, removes from the list of vertices the ones partially covered by the selected descriptor, and proceeds with the next best match, until no vertex remains over the target. We then register the patches corresponding to each matched pair and reject the ones that do not satisfy the distance threshold ϵ. Figure 4 visualizes the candidate patches obtained with this procedure at multiple scales.

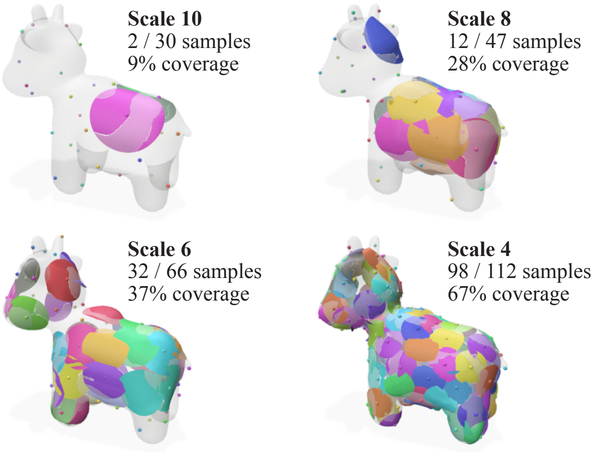
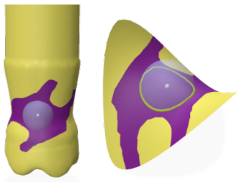


Figure 4: *We generate candidate panels by matching descriptors at multiple scales. While the coarse scales provide large panels, few of them are fully within threshold distance (top). Finer scales provide better coverage (bottom), but the corresponding panels might not extend much beyond their initial disc shape.*

Note that while our descriptor is computed over compact, disc-shaped patches around a point, the resulting registration f provides a transformation aligning the entire source to the target shape at this point. This means that within a given distance threshold, there is the potential to extend patches beyond the best-fitting spherical extent (shown in purple in the inset). The optimization described next operates on these maximally enlarged candidate panels.



Partitioning of the target. Let $\mathcal{F}$ be the set of candidate registrations of the source given by the seeding procedure. Our goal is now to select a small subset $\{f_c\} \subset \mathcal{F}$ of these registrations, together with a panel $\mathcal{S}_c$ associated with each registration, such that the transformed panels cover the target well. Following prior work on partial shape registration [CZ08] and shape approximation [CSaLM13, IRHS20, HMA15, YAB*22], we cast this selection as a graphcut problem by associating a label l_r to each candidate registration $f_r \in \mathcal{F}$. The selection then consists in assigning one such label to each vertex $v \in \mathcal{V_T}$ of the *target* mesh, which we denote as the mapping $l : \mathcal{V_T} \rightarrow \{l_r\}$. We measure the quality of a given labeling l by reformulating Equation 1:

$$\begin{aligned}\mathcal{L}_{\text{partition}}(l) = \mathcal{L}_{\text{accuracy}}(l) \\ + w_{\text{smoothness}}\mathcal{L}_{\text{smoothness}}(l) \\ + w_{\text{fragmentation}}\mathcal{L}_{\text{fragmentation}}(l).\end{aligned} \tag{4}$$

We solve for the labeling that minimizes this objective function using an off-the-shelf solver [DOIB12][†].

Accuracy of the reconstruction. We measure the accuracy of a registration f_r for a given target vertex v as the distance between v and the closest point on the transformed source $f_r(\mathcal{S})$. Since we want to stay within a user-specified tolerance, we map this distance to a high value T when it exceeds the threshold ϵ, and include a special label l_ϵ with a constant cost of ϵ so that the optimization selects this label when no candidate falls within threshold:

$$d_\epsilon(p,q) = \begin{cases} \|p-q\| & \text{if } \|p-q\| \leq \epsilon\,, \\ T & \text{otherwise} \end{cases} \tag{5}$$

$$\mathcal{L}_{\text{accuracy}}(l) = \sum_{v \in \mathcal{V_T}} \min_{p \in f_{l(v)}(\mathcal{S})} d_\epsilon(p,v). \tag{6}$$

In our implementation, we set T to be an order of magnitude more than the highest observed value in our data term, i.e:

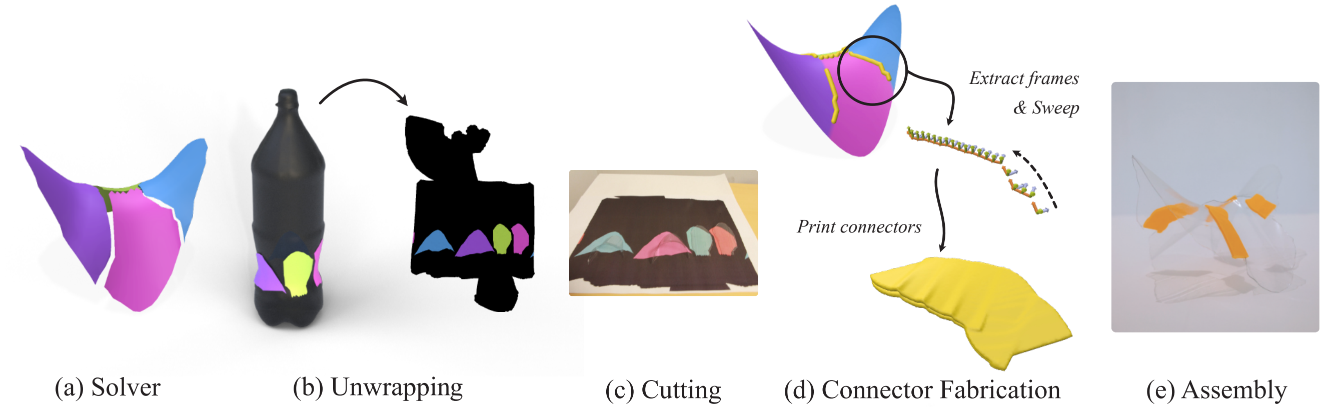
$$T = 10 \max_{v \in \mathcal{V_T}} \min_{p \in f_{l(v)}(\mathcal{S})} d_\epsilon(p,v)$$

Simplicity of the panels. We favor panels with smooth boundaries using a standard term that penalizes changes of labels between vertices connected by an edge in the mesh [BJ01]:

$$\mathcal{L}_{\text{smoothness}}(l) = \sum_{\{u,v\} \in \mathcal{E_T}} 1 - \delta(l(u), l(v)), \tag{7}$$

where $\mathcal{E_T}$ denotes the set of edges of the target mesh and δ is the Kronecker function, $\delta(i,j) = 1$ if $i = j$, $\delta(i,j) = 0$ otherwise.

[†] https://vision.cs.uwaterloo.ca/code/



(a) Solver (b) Unwrapping (c) Cutting (d) Connector Fabrication (e) Assembly

Figure 5: *Given the partition produced by our algorithm (a), we unwrap the plastic bottle to print a cutting pattern (b,c). For each pair of adjacent panels, we generate connectors with thin slits by sweeping a H-profile along the boundary curve (d). Finally, we assemble the target by connecting the cut panels (e).*

Fragmentation. While the smoothness term described above tends to favor the emergence of a small number of large panels, we complement this term with a third objective that aims at simplifying fabrication by explicitly minimizing the number of panels to cut and assemble, which we measure as the number of labels used in the solution [YAB*22, DOIB12]:

$$\mathcal{L}_{\text{fragmentation}}(l) = \sum_{l_r \in \{l(v) | v \in \mathcal{V}_{\mathcal{T}}\}} 1. \quad (8)$$

Note that the number of labels used is only a lower bound on the number of panels that form the reconstruction, since several panels might correspond to disconnected components sharing the same label. Fortunately, such cases are rare in practice for the curved shapes we considered.

Projection on the source. The labeling partitions the target into panels $\{\mathcal{T}_c\}$. The last step of our method consists in extracting the boundary of each panel and projecting these closed curves onto the source according to f_c^{-1} to obtain the source panels $\{\mathcal{S}_c = f_c^{-1}(\mathcal{T}_c)\}$ that we need to cut. We rely on Blender's mesh boolean implementation to perform this cutting operation.

4. Shape Fabrication by Reuse

Our algorithm can apply to diverse manufacturing scenarios, ranging from reuse of small household objects for rapid prototyping to reuse of large structures such as aircraft fuselages or freeform building facades, as illustrated in Section 5. We now describe our initial experiments with the former scenario, focusing on reuse of plastic bottles as a means to fabricate lightweight prototypes of freeform shapes. Figure 5 illustrates the main steps of this process.

Cutting pattern. Our partitioning algorithm produces a set of panels $\{\mathcal{S}_c\}$ that need to be cut from the source shape. Since we do not prevent overlap of these panels on the source, several copies of the source might be needed for manufacturing. In the case of plastic bottles, we leverage rotational symmetry to manually reduce overlap by sliding the panels around the parallels of the bottle, a task that could be automated in the future to achieve optimal packing. We then utilize Boundary-First Flattening [SC17] to produce a map of the bottles containing our source patches. We print and wrap this map around our physical bottles and cut the parts by hand.

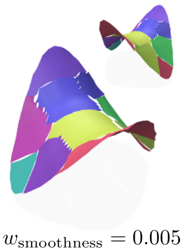
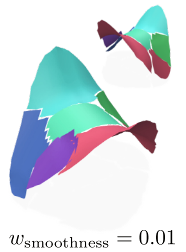
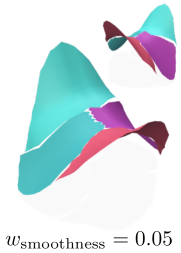
While our formulation favors few, large panels, nothing prevents the presence of spurious small panels. We ignore these panels when they are too difficult to assemble (less than a centimeter wide).

Connectors. To assemble the target, we fabricate custom 3D-printed connectors that join adjacent pairs of panels along their shared boundary. We first compute the Darboux frame at each vertex of the target boundary curve, which we construct by interpolating the normal of adjacent mesh vertices and assigning an arbitrary orientation to the curve. This sequence of frames gives us the trajectory that the connector should follow on the target surface. We generate the connector by sweeping an H-profile along this trajectory [SAJ21] to obtain one slit on each side of the curve, into which we slide the panels. However, the transformed source panels might not meet perfectly along the target boundary curve after registration. We detect such cases by computing the distance of each boundary curve vertex to the two transformed source panels it separates, and only keep the portions of the curve for which these distances are below a threshold. We accommodate for the remaining distance by leveraging the flexibility of the plastic material to bend the panels such that they fit in the connector slits. We print the resulting connectors using an ordinary fused-filament 3D printer.

5. Results

Our formulation balances the accuracy of reconstruction with the simplicity and number of panels. Figure 6 illustrates this tradeoff on a simple saddle shape, where increasing the weight of the smoothness term yields fewer panels with simpler boundaries, at the cost of higher deviation from the target. In contrast, lowering the distance threshold ϵ results in a greater number of panels (Figure 7).

Figure 9 provides a gallery of target shapes approximated from diverse sources. When necessary, we re-meshed these objects such

$w_{\text{smoothness}} = 0.005$ $L_{\text{accuracy}} = 0.00029$ 11 Patches

$w_{\text{smoothness}} = 0.01$ $L_{\text{accuracy}} = 0.00047$ 7 Patches

$w_{\text{smoothness}} = 0.05$ $L_{\text{accuracy}} = 0.00074$ 4 Patches

Figure 6: *Increasing the weight $w_{\text{smoothness}}$ penalizes intricate boundaries and, in combination with the fragmentation term, promotes the emergence of a few, large panels. However, this simplification comes at the price of a higher distance to the target.*

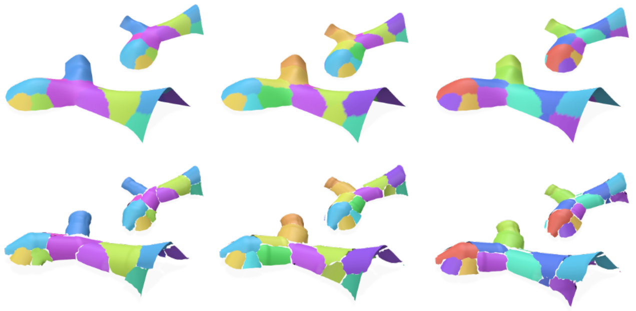


Figure 7: *Lowering threshold ϵ partitions the shape into a greater number of panels to achieve coverage within tolerance. Partitioned target (top) and reconstruction (bottom).*

that the source and target surfaces have a relatively uniform density of vertices. The first row illustrates a rapid prototyping scenario where the *Spot* model [CPS13] is approximated by 23 panels cut from big and small plastic bottles. The second row shows reuse of a large architectural shape to approximate the roof of the Lilium tower. Finally, the third row illustrates a potential reuse scenario for the aeronautic industry. Table 1 provides detailed statistics for each result in terms of parameters used, number of panels, and computation time.

The choices for min and max sphere radii $r_{\min}$, $r_{\max}$, and number of scales K are influenced by several factors. As our balls are extrinsic, setting $r_{\max}$ too high will yield many rejected patches with non-disk topology (especially on long narrow shapes). Setting $r_{\min}$ too low or K too high will increase the time spent seeding the graph cut with extremely tiny patches. In practice we experimentally set $r_{\min}$ to cover at least a single vertex ring on our target, and $r_{\max}$ to be smaller than the smallest axis of the source and target bounding boxes. We pick $w_{\text{smoothness}}$ and $w_{\text{fragmentation}}$ experimentally from a coarse line search, observing that changes to these parameters usually have discontinuous effects on the number of resulting patches and coverage percentage. All the models in Figure 9 have >99.9% coverage. The threshold ϵ is chosen as a fraction of the target shape bounding box diagonal (Under the assumption that larger shapes permit more absolute deviation), by default set to 2%.

Figure 1 and 8 show prototypes we fabricated by reusing plastic bottles. We created virtual models of these bottles from measurements of their diameters at different heights, future work might rely on 3D scanners for this purpose. While the saddle is made of only 4 panels, the architectural shape contains 16 panels and is 44 cm wide. Metrics are summarized in Table 2.

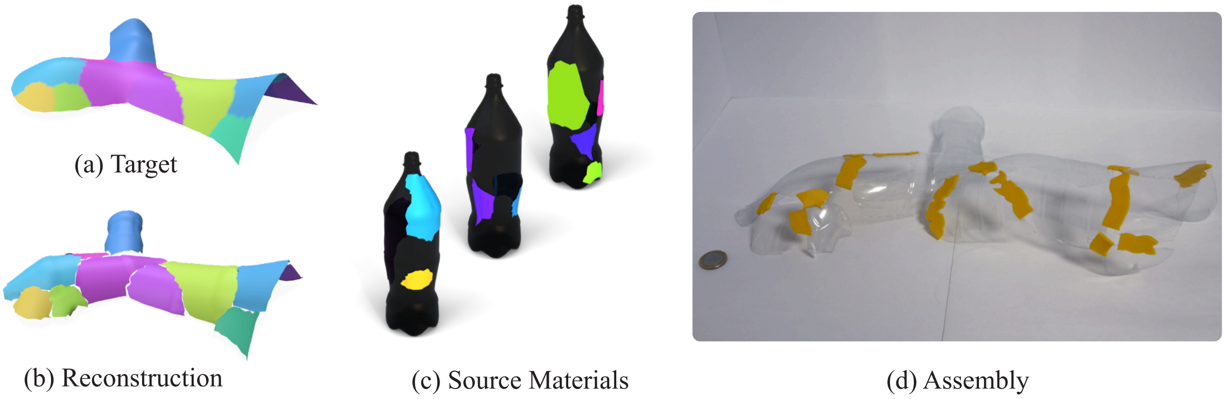
Limitations. While we have demonstrated the potential of surface reuse to approximate freeform shapes, our approach has limitations that point to interesting directions for future work.

To achieve a good tradeoff between accuracy and complexity, the source shape needs to contain similar panels to the target. For the results shown in this paper, we manually selected promising pairs of shapes that have similar mean and Gaussian curvature histograms. In the future, our approach would benefit from shape retrieval methods that could suggest source shapes having similar parts to a given target.

Our work focuses on approximating the overall shape of the target. Complementary objectives could be considered in the future, such as the structural strength of the resulting assembly. To reduce waste, additional terms could be considered to penalize multiple reuse of the same part of the source, or to achieve efficient packing of the panels.

Our mathematical formulation measures the overall accuracy of the reconstruction, but it does not measure whether neighboring panels connect seamlessly. As a result, adjacent panels can lie at a different distance to the target, resulting in gaps of at most 2ϵ width. Fortunately, our application to rapid prototyping does not suffer from this limitation because plastic panels are sufficiently flexible to accommodate small gaps. Nevertheless, we do not model this flexibility explicitly, and panels might be difficult to assemble when competing gaps occur along their boundaries. A possible solution could be to replace our graphcut formulation by a binary programming formulation that would select intersecting portions of the registered candidates to form a manifold surface close to the target, as has been done for planar approximations [NW17]. However, in addition to being computationally expensive, such an approach would require computing the partition of space formed by intersecting registered sources, a non-trivial task [CLSA20]. Another limitation of our current geometry processing pipeline is that we extract the panels from the source mesh by re-projecting the panel boundaries computed over the target, which might fail in the presence of thin structures where consecutive vertices of the boundary can project on disconnected parts of the shape.

Finally, our current fabrication pipeline serves as a proof of concept for building small lightweight prototypes, but suffers from several approximations. In particular, we do not take into account material properties, such as the thickness and flexibility of the panels that can vary for different types of plastic bottles. Our solution of flattening and printing the cutting patterns on paper to be wrapped around bottles is also approximate, higher precision might


(a) Target

(b) Reconstruction

(c) Source Materials

(d) Assembly

Figure 8: *Constructing an architectural model using our method. The solver partitions the panels (a,b) that we project upon a series of bottles (c). After cutting these bottles and printing the connectors, we assemble the shape (d).*

Source	V_S	Target	V_T	ϵ	$w_{\text{smoothness}}$	$w_{\text{fragmentation}}$	K	P	L_{accuracy}	Desc (s)	Seeding (s)	GC (s)
Airplane	8519	Car	1131	0.04	0.050	0.050	8	9	0.011	2.2	98.3	6.5
Architecture	3551	Lilium	3389	0.40	0.025	0.025	8	19	0.070	2.1	238.6	28.6
Bottles	7337	Cow	2930	0.01	0.020	0.100	5	23	0.001	3.2	7.6	7.5

Table 1: *Parameters & metrics for the results of Figure 9, where ϵ denotes the distance threshold set by the user and K denotes the number of scales used during sampling. The primary quality metrics are the number of patches P and the chamfer distance of the patches to the surface L_{accuracy}. The last columns provide the duration in seconds for each stage of our pipeline: descriptor computation, patch seeding, and graphcut.*

Source	*Target*	*P*	*C*	*Dims*	*# Bottles*
Bottles	Architecture	12	16	44 x 9 x 32	4L + 1S
Bottles	Saddle	4	4	10 x 8 x 11	1L

Table 2: *Metrics for our physical results, as shown in Figure 1 and Figure 8. P denotes the number of patches used, and C the number of connectors. Dimensions of the Target are expressed in cm (width, depth, height). L/S denote the number of large and small bottles required for fabrication respectively.*

be achieved with CNC machining. For these reasons, our pipeline is better suited to the fabrication of approximate models, for instance in architecture where prototypes are often assembled by reusing foam, cardboard and plastic [Dun14]. From a circular design perspective, while our approach offers a second life to plastic bottles, assembling bottle panels with 3D-printed connectors might hinder subsequent recycling, as the different types of plastic might need to be separated. Additionally, while our partitioning algorithm can handle surface discontinuities, our procedure to generate connectors currently assumes that neighboring panels connect continuously at the H-profile that we sweep along their shared boundary. Sharp twisting along boundary curves is also problematic for fabrication, as it can lead to “clogging” along the slits of the H-profile as the material is being deposited. This issue requires manual post-print cleanup and can cause printing failure.

6. Conclusion

The development of CAD-CAM has largely been driven by the linear production model, offering designers an extensive toolset to create shapes that are manufactured by processing raw materials, for instance through milling, molding, bending. In contrast, we explored a circular production model that reduces material consumption and processing by reusing parts of discarded products, but requires designers to work within the limits of prescribed source shapes. Motivated by this emerging workflow, we have proposed a mathematical formulation of surface approximation by reuse, along with an algorithm to best reproduce a target shape from a given source. We hope that this work will inspire further research on computer-aided circular design, in particular to augment our formulation with additional constraints on panel connectivity to ensure that cutting and assembly can be performed at a larger scale than the prototypes we have produced.

Acknowledgments

We thank Marzia Riso for help with pictures and video, Jelle Joustra and Mariana Popescu for inspirational discussions on structural reuse, and the reviewers for their kind and constructive feedback. We also thank the developers of the libraries used for implementation: Polyscope [S*19], Geometry Central [SC*19], Gpytoolbox [SS*23] and libigl [JP*18]. This research was supported by the Inria CircleD Exploratory Action, which benefits from the support of the La Poste Group, patron of the Inria Foundation, and by the TU Delft Climate Action Funding and the TU Delft France Initiative.

References

[AHM*20] Amtsberg F., Huang Y., Marshall D. J. M., Moreno K., Mueller C.: Structural Up-cycling: Matching Digital and Natural Geometry. In *Advances in Architectural Geometry (AAG)* (2020). 1, 2

[BB08] Bronstein A. M., Bronstein M. M.: Regularized Partial Matching of Rigid Shapes. In *Computer Vision (ECCV)*. 2008. doi:10.1007/978-3-540-88688-4_11. 3

[BDSF19] Brütting J., Desruelle J., Senatore G., Fivet C.: Design of Truss Structures Through Reuse. *Structures 18* (Apr. 2019). doi:10.1016/j.istruc.2018.11.006. 1, 2

[BdvZ14] Bakker C., den Hollander M., van Hinte E., Zijlstra Y.: *Products that last: Product design for circular business models*. Bis Publisher, 2014. 1

[BJ01] Boykov Y., Jolly M.-P.: Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *Proc. IEEE International Conference on Computer Vision (ICCV)* (2001), vol. 1. doi:10.1109/ICCV.2001.937505. 5

[BVHSH21] Binninger A., Verhoeven F., Herholz P., Sorkine-Hornung O.: Developable Approximation via Gauss Image Thinning. *Computer Graphics Forum* (2021). doi:10.1111/cgf.14374. 2

[CLSA20] Cherchi G., Livesu M., Scateni R., Attene M.: Fast and robust mesh arrangements using floating-point arithmetic. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 39*, 6 (2020). doi:10.1145/3414685.3417818. 7

[CPS13] Crane K., Pinkall U., Schröder P.: Robust fairing via conformal curvature flow. *ACM Transactions on Graphics (TOG) 32*, 4 (2013). doi:10.1145/2461912.2461986. 7

[CQS*23] Chen R., Qiu P., Song P., Deng B., Wang Z., He Y.: Masonry shell structures with discrete equivalence classes. *ACM Transactions on Graphics (Proc. SIGGRAPH) 42*, 4 (2023). doi:10.1145/3592095. 2

[CSaLM13] Chen D., Sitthi-amorn P., Lan J. T., Matusik W.: Computing and Fabricating Multiplanar Models. *Computer Graphics Forum (Proc. Eurographics)* (2013). doi:10.1111/cgf.12050. 2, 5

[CZ08] Chang W., Zwicker M.: Automatic registration for articulated shapes. *Computer Graphics Forum (Proc. SGP) 27*, 5 (2008). doi:10.1111/j.1467-8659.2008.01286.x. 1, 3, 5

[DOIB12] Delong A., Osokin A., Isack H. N., Boykov Y.: Fast approximate energy minimization with label costs. *International Journal of Computer Vision 96* (2012). doi:10.1109/CVPR.2010.5539897. 2, 3, 5, 6

[Dun14] Dunn N.: *Architectural Modelmaking (second edition)*. Laurence King Publishing, 2014. 8

[DYDZ22] Deng B., Yao Y., Dyke R. M., Zhang J.: A survey of non-rigid 3d registration. *Computer Graphics Forum 41*, 2 (2022). doi:10.1111/cgf.14502. 3

[DYYT17] Duncan N., Yu L.-F., Yeung S.-K., Terzopoulos D.: Approximate dissections. *ACM Transactions on Graphics 36*, 6 (Dec. 2017). doi:10.1145/3130800.3130831. 2

[EKS*10] Eigensatz M., Kilian M., Schiftner A., Mitra N. J., Pottmann H., Pauly M.: Paneling architectural freeform surfaces. *ACM Transactions on Graphics 29*, 4 (Jul. 2010). doi:10.1145/1778765.1778782. 2

[FB20] Fivet C., Brütting J.: Nothing is lost, nothing is created, everything is reused: structural design for a circular economy. *Structural Engineer 98* (01 2020). doi:10.56330/LXAH1188. 1

[FKS*04] Funkhouser T., Kazhdan M., Shilane P., Min P., Kiefer W., Tal A., Rusinkiewicz S., Dobkin D.: Modeling by example. *SIGGRAPH Conference Papers* (2004). doi:10.1145/1186562.1015775. 2

[FLC*25] Favilli A., Laccone F., Cignoni P., Malomo L., Giorgi D.: Optimizing free-form grid shells with reclaimed elements under inventory constraints. *Computer Graphics Forum (Proc. Eurographics)* (2025). doi:10.1111/cgf.70047. 1, 2

[FLHCO10] Fu C.-W., Lai C.-F., He Y., Cohen-Or D.: K-set tilable surfaces. *ACM Transactions on Graphics (Proc. SIGGRAPH) 29*, 4 (2010). doi:10.1145/1778765.1778781. 2

[GBK16] Gehre A., Bommes D., Kobbelt L.: Geodesic Iso-Curve Signature. In *Proc. of the Conference on Vision, Modeling and Visualization (VMV)* (2016). 4

[GSP*07] Gal R., Sorkine O., Popa T., Sheffer A., Cohen-Or D.: 3D collage: Expressive non-realistic modeling. In *Proc. International Symposium on Non-Photorealistic Animation and Rendering (NPAR)* (2007). doi:10.1145/1274871.1274873. 2

[HADWM21] Huang Y., Alkhayat L., De Wolf C., Mueller C.: Algorithmic circular design with reused structural elements: method and tool. doi:1721.1/145562. 2

[HMA15] Herholz P., Matusik W., Alexa M.: Approximating free-form geometry with height fields for manufacturing. *Computer Graphics Forum (Proc. Eurographics) 34*, 2 (2015). doi:10.1111/cgf.12556. 2, 5

[HPPLG11] Heider P., Pierre-Pierre A., Li R., Grimm C.: Local Shape Descriptors, a Survey and Evaluation. In *Eurographics Workshop on 3D Object Retrieval (3DOR)* (2011). doi:10.2312/3DOR/3DOR11/049-056. 4

[IRHS20] Ion A., Rabinovich M., Herholz P., Sorkine-Hornung O.: Shape approximation by developable wrapping. *ACM Transactions on Graphics (Proc. SIGGRAPH ASIA) 39*, 6 (2020). doi:10.1145/3414685.3417835. 2, 5

[JFB21] Joustra J., Flipsen B., Balkenende R.: Structural reuse of wind turbine blades through segmentation. *Composites Part C: Open Access 5* (2021). doi:10.1016/j.jcomc.2021.100137. 1, 2

[JP*18] Jacobson A., Panozzo D., et al.: libigl: A simple C++ geometry processing library, 2018. URL: https://libigl.github.io/. 8

[JSVB22] Jourdan D., Skouras M., Vouga E., Bousseau A.: Computational design of self-actuated surfaces by printing plastic ribbons on stretched fabric. *Computer Graphics Forum (Proc. Eurographics) 41*, 2 (2022). doi:10.1111/cgf.14489. 2

[KHLM16] Koo B., Hergel J., Lefebvre S., Mitra N. J.: Towards zero-waste furniture design. *IEEE Transactions on Visualization and Computer Graphics (TVCG) 23*, 12 (2016). doi:10.1109/TVCG.2016.2633519. 2

[KSW*17] Kovacs R., Seufert A., Wall L., Chen H.-T., Meinel F., Müller W., You S., Brehm M., Striebel J., Kommana Y., Popiak A., Bläsius T., Baudisch P.: TrussFab: Fabricating Sturdy Large-Scale Structures on Desktop 3D Printers. In *Proc. ACM Conference on Human Factors in Computing Systems (CHI)* (May 2017). doi:10.1145/3025453.3026016. 2

[LB17] Liu P., Barlow C. Y.: Wind turbine blade waste in 2050. *Waste Management 62* (2017). doi:10.1016/j.wasman.2017.02.007. 2

[LBB12] Litany O., Bronstein A. M., Bronstein M. M.: Putting the Pieces Together: Regularized Multi-part Shape Matching. In *Computer Vision (ECCV)*. 2012. doi:10.1007/978-3-642-33863-2_1. 2

[LGJ*17] Lefeuvre A., Garnier S., Jacquemin L., Pillain B., Sonnemann G.: Anticipating in-use stocks of carbon fiber reinforced polymers and related waste flows generated by the commercial aeronautical sector until 2050. *Resources, Conservation and Recycling 125* (2017). doi:10.1016/j.resconrec.2017.06.023. 2

Authors version. Published in Computer Graphics Forum (SGP 2025)

[LMAH*18] Li S., Mahdavi-Amiri A., Hu R., Liu H., Zou C., Van Kaick O., Liu X., Huang H., Zhang H.: Construction and fabrication of reversible shape transforms. *ACM Transactions on Graphics (TOG) 37*, 6 (Dec. 2018). doi:10.1145/3272127.3275061. 2

[LZW*25] Li H., Zhang N., Wang L., Lu J.-X., Dong R., Duan H., Yang J.: The challenge of recycling fast-growing fibre-reinforced polymer waste. *Nature Reviews Materials* (Jan 2025). doi:10.1038/s41578-024-00762-2. 2

[MGP06] Mitra N. J., Guibas L., Pauly M.: Partial and approximate symmetry detection for 3d geometry. *ACM Transactions on Graphics (Proc. SIGGRAPH) 25*, 3 (2006). doi:10.1145/1141911.1141924. 1, 3

[MJC*24] Mei Y., Jones B., Cascaval D., Mankoff J., Vouga E., Schulz A.: Fabhacks: Transform everyday objects into home hacks leveraging a solver-aided dsl. In *Proc. ACM Symposium on Computational Fabrication (SCF)* (2024). doi:10.1145/3639473.3665788. 2

[MPS*04] Mortara M., Patané G., Spagnuolo M., Falcidieno B., Rossignac J.: Blowing Bubbles for Multi-Scale Analysis and Decomposition of Triangle Meshes. *Algorithmica 38*, 1 (Jan. 2004). doi:10.1007/s00453-003-1051-4. 4

[MS16] Mollica Z., Self M.: Tree fork truss: geometric strategies for exploiting inherent material form. In *Advances in Architectural Geometry* (2016). doi:10.3218/3778-4_11. 1, 2

[NW17] Nan L., Wonka P.: Polyfit: Polygonal surface reconstruction from point clouds. In *2017 IEEE International Conference on Computer Vision (ICCV)* (2017). doi:10.1109/ICCV.2017.258. 7

[PIC*21] Panetta J., Isvoranu F., Chen T., Siéfert E., Roman B., Pauly M.: Computational inverse design of surface-based inflatables. *ACM Transactions on Graphics (Proc. SIGGRAPH) 40*, 4 (2021). doi:10.1145/3450626.3459789. 2

[PWHY09] Pottmann H., Wallner J., Huang Q.-X., Yang Y.-L.: Integral invariants for robust geometry processing. *Computer Aided Geometric Design 26*, 1 (2009). doi:10.1016/j.cagd.2008.01.002. 4

[QPK*25] Qi A., Pietroni N., Korosteleva M., Sorkine-Hornung O., Bousseau A.: Rags2riches: Computational garment reuse. doi:10.1145/3721238.3730703. 2

[S*19] Sharp N., et al.: Polyscope, 2019. URL: www.polyscope.run. 8

[SAJ20] Sellán S., Aigerman N., Jacobson A.: Developability of heightfields via rank minimization. *ACM Transactions on Graphics* (2020). doi:10.1145/3386569.3392419. 2

[SAJ21] Sellán S., Aigerman N., Jacobson A.: Swept volumes via spacetime numerical continuation. *ACM Transactions on Graphics (TOG) 40*, 4 (2021). doi:10.1145/3450626.3459780. 6

[SC17] Sawhney R., Crane K.: Boundary first flattening. *ACM Transactions on Graphics (TOG) 37*, 1 (2017). doi:10.1145/3132705. 6

[SC*19] Sharp N., Crane K., et al.: Geometrycentral: A modern c++ library of data structures and algorithms for geometry processing, 2019. URL: https://geometry-central.net/. 8

[SFCH12] Shen C.-H., Fu H., Chen K., Hu S.-M.: Structure recovery by part assembly. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 31*, 6 (Nov. 2012). doi:10.1145/2366145.2366199. 2

[SFJ*17] Song P., Fu C.-W., Jin Y., Xu H., Liu L., Heng P.-A., Cohen-Or D.: Reconfigurable interlocking furniture. *ACM Transactions on Graphics (TOG) 36*, 6 (Nov. 2017). doi:10.1145/3130800.3130803. 2

[SGC18] Stein O., Grinspun E., Crane K.: Developability of triangle meshes. *ACM Transactions on Graphics (Proc. SIGGRAPH) 37*, 4 (2018). doi:10.1145/3197517.3201303. 2

[SOG09] Sun J., Ovsjanikov M., Guibas L.: A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. *Computer Graphics Forum 28*, 5 (July 2009). doi:10.1111/j.1467-8659.2009.01515.x. 4

[SRVSH24] Segall A., Ren J., Vaxman A., Sorkine-Hornung O.: Fabric tessellation: Realizing freeform surfaces by smocking. *ACM Transaction on Graphics (Proc. SIGGRAPH) 43*, 4 (2024). doi:10.1145/3658151. 2

[SS10] Singh M., Schaefer S.: Triangle surfaces with discrete equivalence classes. *ACM Transactions on Graphics (Proc. SIGGRAPH) 29*, 4 (2010). doi:10.1145/1778765.1778783. 2

[SS*23] Sellán S., Stein O., et al.: gptoolbox: A python geometry processing toolbox, 2023. URL: https://gpytoolbox.org/. 8

[THL23] Tomczak A., Haakonsen S. M., Luczkowski M.: Matching algorithms to assist in designing with reclaimed building elements. *Environmental Research: Infrastructure and Sustainability 3*, 3 (Sept. 2023). Publisher: IOP Publishing. doi:10.1088/2634-4505/acf341. 2

[TSW*19] Tang K., Song P., Wang X., Deng B., Fu C., Liu L.: Computational Design of Steady 3D Dissection Puzzles. *Computer Graphics Forum 38*, 2 (May 2019). doi:10.1111/cgf.13638. 2

[vKZHCO11] van Kaick O., Zhang H., Hamarneh G., Cohen-Or D.: A survey on shape correspondence. *Computer Graphics Forum 30*, 6 (2011). doi:10.1111/j.1467-8659.2011.01884.x. 3, 4

[VMLC24] Van Marcke A., Laghi V., Carstensen J. V.: Automated planar truss design with reclaimed partially disassembled steel truss components. *Journal of Building Engineering 84* (May 2024). doi:10.1016/j.jobe.2024.108458. 1, 2

[WJVS21] Wall L. W., Jacobson A., Vogel D., Schneider O.: Scrappy: Using scrap material as infill to make fabrication more sustainable. In *Proc. ACM Conference on Human Factors in Computing Systems (CHI)* (2021). doi:10.1145/3411764.3445187. 2

[WSP21] Wang Z., Song P., Pauly M.: State of the art on computational design of assemblies with rigid parts. *Computer Graphics Forum 40*, 2 (2021). doi:10.1111/cgf.142660. 3

[XCH*24] Xu Q.-C., Chen H.-X., Hua J., Zhan X., Yang Y.-L., Mu T.-J.: Fragmentdiff: A diffusion model for fractured object assembly. In *SIGGRAPH Asia Conference Papers* (2024). doi:10.1145/3680528.3687673. 2

[YAB*22] Yu E., Arora R., Bærentzen J. A., Singh K., Bousseau A.: Piecewise-smooth surface fitting onto unstructured 3d sketches. *ACM Transactions on Graphics (Proc. SIGGRAPH)* (2022). doi:10.1145/3528223.3530100. 2, 5, 6

[YLI19] Yoshida H., Larsson M., Igarashi T.: Upcycling Tree Branches as Architectural Elements through Collaborative Design and Fabrication. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI)* (Mar. 2019), ACM. doi:10.1145/3294109.3295639. 2

[ZMB*24] Zhang R., Mueller S., Bernstein G. L., Schulz A., Leake M.: Wastebanned: Supporting zero waste fashion design through linked edits. In *Proc. ACM Symposium on User Interface Software and Technology (UIST)* (2024). doi:10.1145/3654777.3676395. 2

[ZW12] Zhou Y., Wang R.: An Algorithm for Creating Geometric Dissection Puzzles. 2

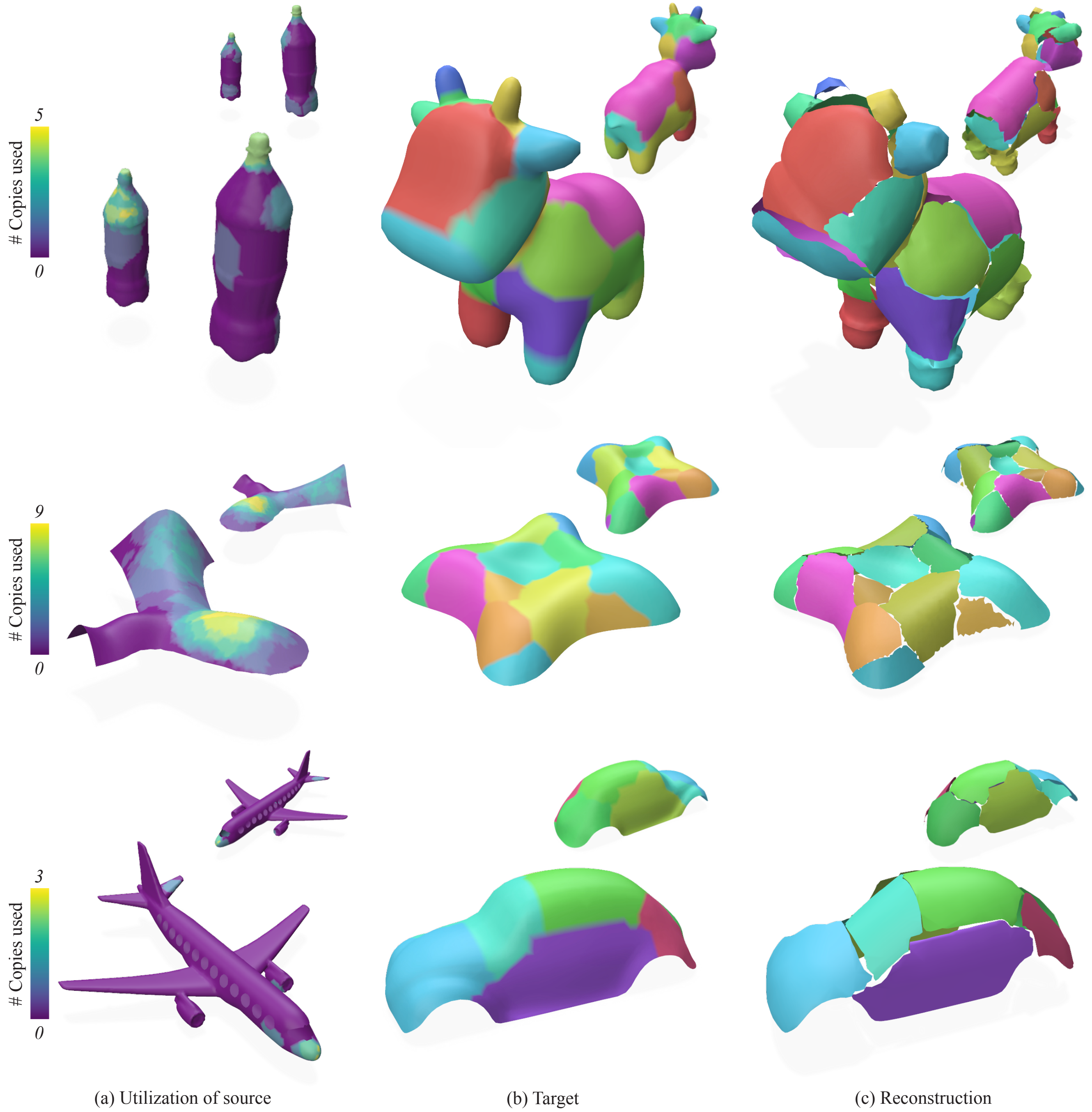


Figure 9: *For each result, we show the source shape with a heatmap to indicate which parts are reused and how often (a), along with the partition of the target (b) and its reconstruction by the transformed panels cut from the source (c). We show a second viewpoint on source and target as inset.*

Authors version. Published in Computer Graphics Forum (SGP 2025)