Synthesis of 3D Jigsaw Puzzles over Freeform 2-Manifolds

Gershon Elber
Dept. of Computer Science
Technion, Israel Inst. of Technology
Haifa 32000, Israel

Myung-Soo Kim
Dept. of Computer Engineering
Seoul National University
Seoul 151-742, South Korea

Abstract

We present a simple algorithm for synthesizing 3D jigsaw puzzles from arbitrary 3D freeform 2-manifold geometric models represented with trimmed NURBS surfaces. The construction algorithm is based on a few conventional geometric operations on freeform curves and surfaces. In particular, we need to compute the offset of freeform NURBS surfaces (for thickening the 2-manifold surfaces) and the functional composition of a univariate curve representation to a bivariate rational surface (for breaking up a 3D model into curved jigsaw tiles). It is thus almost straightforward to convert the proposed algorithm to a practical system using standard tools available in B-rep based geometric modeling systems, that employ trimmed NURBS surfaces. We demonstrate the effectiveness of the proposed approach by fabricating several test sets of 3D jigsaw puzzles for freeform solids consisting of trimmed NURBS surface models.

1 Introduction

Jigsaw puzzles are the most popular puzzles beloved by juniors and seniors alike [5]. Even small children can enjoy playing with jigsaw puzzles of a few dozens of pieces. Jigsaw puzzles improve the visual-spatial reasoning power of the human brain. There are many surprising benefits in assembling jigsaw puzzles, for all ages, and in particular, for dementia patients by improving brain functions, in short-term memory. Motivated with these good reasons for promoting the computational art, we aim to develop support of classic jigsaw puzzles that will be more general than the state-of-the-art, and in 3D, in specific. A significant portion of industrial artifacts are nowadays geometrically designed with and manufactured from 3D NURBS models [6, 9, 12]. In this paper, we propose a practical solution for converting freeform (possibly trimmed) surface models to 3D jigsaw puzzles that can be fabricated using 3D printers.

As shown in Figure 1, the contemporary state-of-the-art 3D jigsaw puzzles are mostly composed of jigsaw pieces placed along planar or developable surface panels. (Developable surfaces include cylinders, cones, and those obtained by bending planes without stretching or compression.) There are certainly some technical difficulties that must be resolved, in order to extend the current plane-based method to a general doubly-curved, possibly trimmed, surface-based technology.

Much of geometric industrial design yields products that are solids consisting of trimmed NURBS surfaces, where the surface intersection curves delineate the boundary of one surface patch across which the adjoining surface patches are connected in a watertight manner. This common representation, to virtually all geometric modeling systems, is also known as B-reps, for boundary representations. In this work, we deal with the surface trimming problem by generating a sequence of adjoining jigsaw pieces along the trimming curves.

Because our input consists of (possibly trimmed) NURBS surface models, we have full control over the parameterization and aside from singularities, the mapping into tiles is simple. Assuming the given NURBS models are continuous, our construction algorithm transparently converts the trimmed surface models to the desired pieces of 3D jigsaw puzzles. The remaining algorithmic steps are based on conventional algebraic and geometric operations on freeform curves and surfaces such as functional composition of curves on surfaces and the offset computation for curves and surfaces. (They support the fabrication processes for thickening 2-manifold surfaces and breaking up a 3D model into curved jigsaw tiles.) Based on these computational tools, available in most contemporary CAD systems, we can generate fabrication files (in STL or OBJ common file formats) for 3D printing the jigsaw puzzle pieces. Further, the precision of these operations ensure the required accuracy for the fabrication and then the assembly of the component pieces, for a physical realization of the

Figure 1: Conventional 3D jigsaw puzzles mostly composed of planar and/or developable surface pieces.

3D freeform model under consideration.

There are many previous results on the fabrication of 3D puzzles. As we discuss in more details in Section 2, on Previous Work, the majority of these results are developed for the volumetric puzzle representations, including those transformable to other 3D shapes. There are very few exceptions. The 3D polyomino puzzle of Lo et al. [20] is based on a surface representation of quadmesh model, which also requires an offset surface computation for the fabrication of a physical 3D puzzle. The quad-based approach is not directly applicable to trimmed surface models, especially with C^1 discontinuities as well as polar points or singularities. A quadmesh approximation may convert some surface models to those acceptable to the algorithm of Lo et al. [20], but not always, due to certain polyomino constraints on the quadmesh structure.

The main contribution of this work can be summarized as follows:

- A general algorithm is proposed for the synthesis of 3D puzzles for B-rep geometric CAD models.
- The input can be any solid model bounded by trimmed surfaces, including open surfaces.
- The tiles can be of any shape as long as the tiling covers a proper domain, including interlocking tiles.
- C^1-discontinuities in the surface interior and/or along trimming curves are supported.

The rest of this paper is organized as follows. In Section 2, we briefly review related previous results. In Section 3, the different steps of the proposed algorithm are explained using the conventional tools of geometric modeling. In Section 4, several test results are presented, including 3D printed and assembled jigsaw puzzles, only to be concluded in Section 5.

2 Previous Work

Compared with the long history of geometry and mathematics, it is quite surprising that most geometric puzzles have been designed and crafted only within the last 200 years [5]. Jigsaw puzzles are simpler and thus have a longer history (over 200 years, presumably invented by John Spilsbury around 1760) than most other types of geometric puzzles. This also explains the popularity of jigsaw puzzles among the general public and in almost all ages. On the other hand, polyomino puzzles are considerably more involved and even under the subject of recreational mathematics [10].

In computer graphics, the computational design of 3D geometric puzzles has attracted much research attention since the introduction of 3D printing technologies, which made the physical realization of 3D puzzles simpler than ever. (The state-of-the-art techniques currently available on a wide range of fabrication capabilities are reported in Bermano et al. [1] and Bickel et al. [2].) The majority of these recent results are in the form of volumetric dissections [17, 26], often with interlocking structures [16, 27, 28, 32, 35] or even with transformable structures [31, 36, 37].

Sequin [26] used dissection puzzles in a graduate course at UC Berkeley for training the spatial understanding and modeling of solid shapes. Xin et al. [35] proposed an automatic construction algorithm for burr puzzles from 3D mesh models, whereas Song et al. [27] introduced a formal approach to the interlocking mechanism and developed an efficient scheme for constructing decentralized interlocking puzzles. Sun and Zheng [31] introduced dynamically interlocking twisty puzzles that generalize Rubik's Cube and its variants to user-supplied 3D models. Kita and Saito [16] proposed a computational design method for generalizing centrifugal puzzles, by embedding the core mechanisms into complex 3D geometric models.

Compared with the volumetric dissection-based approaches, there are relatively few results on the construction of 3D puzzles on non-planar surfaces. Lo et al. [20] considered the construction of 3D polyomino puzzles on quadmesh models. The quadmesh structure on a non-planar surface introduces additional constraints on the possible topological arrangement of polyomino puzzle pieces.

Consequently, there is a certain limited class of quadmesh models that allow the construction of 3D puzzles.

Lo et al. [20] demonstrated the capability of extending the semiregular patterns (used in the texture mapping of mesh models [14, 15]) to the construction of 3D puzzle pieces, where each piece shape corresponds to a component pattern. In Figure 16, we show the construction of 3D puzzle using an M.C. Escher style tiling. For complex trimmed surface models, possibly after decomposing the trimmed surface patches into rectangular pieces [22], we can apply the semiregular patterns to the resulting 3D model. Tesselating the surfaces into meshes, semi-regular quad meshes can be generated automatically or semi-automatically [3]. The layout of rectangular patches can also be used for the reparameterization of our surface models into a structure well suited for the semiregular texture patterns and the corresponding puzzle pieces.

Assuming the input is a set of trimmed surfaces representing the freeform geometric models (aka B-reps), in this work, we present a transparent translation into the construction of 3D jigsaw puzzle pieces that cover the surface geometry of a given 3D model. The construction algorithm is based on conventional algebraic and geometric operations on freeform curves and surfaces, which are usually provided in contemporary CAD tools for NURBS based geometric modeling. Though we consider the 3D construction of jigsaw puzzles on doubly-curved surfaces, it is also easy to adapt the conventional design techniques for 2D jigsaw puzzles [18].

3 Algorithms

We are now ready to discuss the algorithms behind the 3D jigsaw puzzles. The input is a B-rep model consisting of one or more (trimmed) parametric surfaces. Typically, the model will be closed but it does not have to be. In Section 3.1, we discuss the case of a model consisting of a single tensor product surface. This is extended in Section 3.2 to handle the case of trimmed surfaces and the stitching of the puzzle tiles along the trimming curves.

3.1 A Jigsaw Puzzle of a Single Tensor Product Surface

Let $S(u, v)$, $(u, v \in [0, 1])$, be a parametric surface, typically represented in a Bézier or a B-spline form. S is assumed to be regular, namely, $\frac{\partial S}{\partial u} \times \frac{\partial S}{\partial v}$ never vanishes. While the regularity of the parameterization will help avoiding singular tiles, the parameterization will directly control the placement and size of tiles, as will be shortly revealed. Because

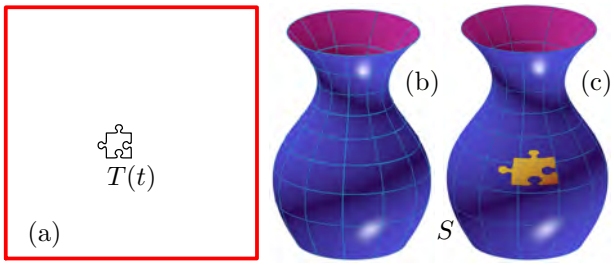

Figure 2: Composing one outline curve of tile $T(t)$ in the domain of S (in (a)), onto S (in (b)), resulting in (c) as $S(T(t))$.

the same surface can have completely different parameterization, the parameterization of a surface is another design degree of freedom of puzzles to consider. With the exception of Section 3.1.5, the input model will be considered given "as is" from the CAD system, and this degree of freedom (surface reparameterization) will not be exploited in this work.

Let $T(t)$ be a jigsaw puzzle tile shape, represented as a closed outline curve, in the domain of S. $T(t) = (t_x(t), t_y(t))$ is assumed to be regular as well, or $\|T'(t)\| \neq 0$.

Now consider the composition of $S(T) = S(t_x(t), t_y(t))$. $S(T)$ is a 3D sub-region of S restricted to the domain in S that is also in the tile $T(t)$. See Figure 2.

Now consider a planar tiling of the whole domain of S. All tiles could be mapped, one by one, to the range of S, yielding a full 3D jigsaw puzzle of the geometry of S. See Figure 3. Note that the top of the vase, in Figure 3 (b), is open and hence the corresponding tiles in the domain, in the top of Figure 3 (a), have no interlocking handles. Further, the left edge of the domain in Figure 3 (a) extends beyond the domain and that should be handled with care, and will be addressed in Section 3.1.3.

While the basic idea is simple, like many other cases, the devil can be found in the details. In the next sections, we examine several concerns that must be addressed for a proper realization of 3D puzzles. Section 3.1.1 discusses the tile shapes that one can use. In Section 3.1.2, the question of tile offset in the plane is discussed, and why it is desired. In Section 3.1.3, the process of extracting the 3D tile shape is portrayed. Then, Section 3.1.4 is handling the addition of realizable (finite) thicknesses to the tiles. Finally, in Section 3.1.5, we consider the possibility of making all tiles a bit different from each other, making it much more challenging to the end user to assemble.

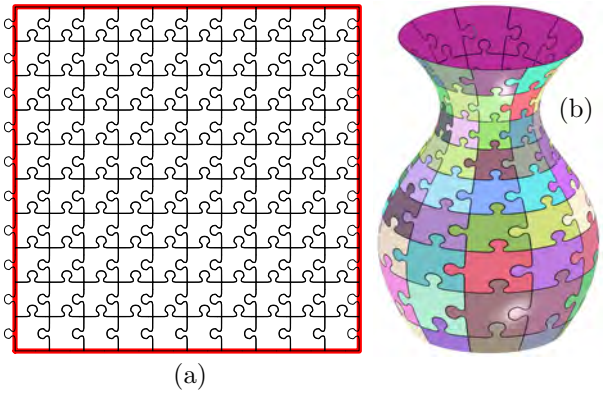

Figure 3: A tiling of the domain of S by puzzle tiles (in (a)), is composed onto S (same as in Figure 2 (b)), resulting in (b). Each tile is uniquely colored.

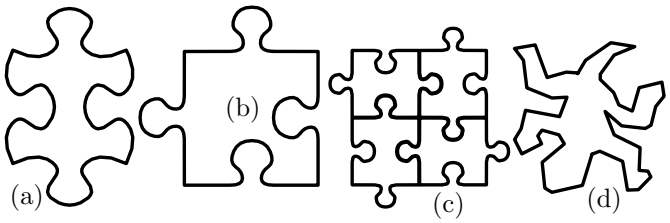

Figure 4: Several simple possibilities of puzzle tiles that can be employed. In (a), a smooth rounded tile is shown. In (b), a more rectangular shape is presented, whereas, in (c), a macro tile is constructed from four tiles of similar shapes. Finally, (d) presents a general tile of M.C. Escher-style, that can also be employed as a puzzle tile.

3.1.1 Selecting the Shape of the Tile

Tilings of the plane by regular, semi-regular shape, isohedral shapes and others is a major topic of research. Herein, we need to add the restriction that adjacent tiles must be interlocking with each other. That said, any tiles that follow this additional restriction can be employed. Possible examples are shown in Figure 4.

The tiles are typically translates of each other. However, in Figure 4 (c), rotations are used to create a *macro-tile* consisting of four elements. This macro element can now be used as an atomic tile undergoing translations only. The same holds for the tile-element in Figure 4 (a), for each four such (rotated) tile-elements can create a translation-only macro-tile. Finally, in Figure 4 (d), an M.C. Escher-style tile is shown that can also be employed in this context. In summary, the tiling of the domains of the surfaces can be established in many possible ways, while we will demonstrate a few such tilings throughout this work.

One concern to handle is proper boundary conditions for S. It is typically undesired that the interlocking handles will terminate on the boundary. However, if S is closed and/or periodic, these interlocking handles must properly cross the periodic boundary to the other side. In other words, the composition of the tiles must properly handle periodic/closed cases. We will return to this concern and show how we manage these handles, in Section 3.1.3. These interlocking handles are clearly seen in Figure 3 (a), leaving the boundary on the left, with the matching holes, on the right. This boundary is the periodic boundary of the surface of revolution of the vase shown in Figure 3 (b). Note that no interlocking handles are at the top nor at the bottom boundaries, in Figure 3 (a), as these are mapped to the top edge and to the bottom (center point) of the vase in Figure 3 (b).

3.1.2 Adding a Planar Offset to the Tile Shape

The tiles are likely to be perfectly aligned in the plane with zero clearance. This precision can be detrimental when aiming to fabricate the tiles and assemble them into a tangible model. Some clearance must be allowed depending on the fabrication process. Two options are available here:

1. The offset of all tiles can be computed in the domain of S, namely, in a plane, or

2. The offset is computed as a geodesic offset over the 3D surface.

The latter has the advantage that it is computed over the actual 3D puzzle tiles. However, computing offsets over surfaces (i.e., the parallel curves on non-Euclidean 2-manifolds) is one of the most challenging problems in geometric modeling [23, 34]. See Elber and Kim [8], for a recent approach to this problem. The first alternative, to compute these offsets in the domain, assumes that the stretch in the mapping of S (magnitudes of the coefficients of the first fundamental form) is bounded. In this work and for simplicity, we chose the former solution of computing the offsets in the domain of S. Figure 5 shows a zoom-in on Figure 3, exposing the offsets employed.

3.1.3 Extracting the Shape of the Puzzle Tile, in 3-space

Given a surface S and the closed planar outline curve of the puzzle tile, $T(t)$, we seek the region of S, in the Euclidean space, that is restricted to $S(T)$. Fortunately, many geometric modeling environments support the construction of *trimmed surfaces* from a tensor product surface S and a

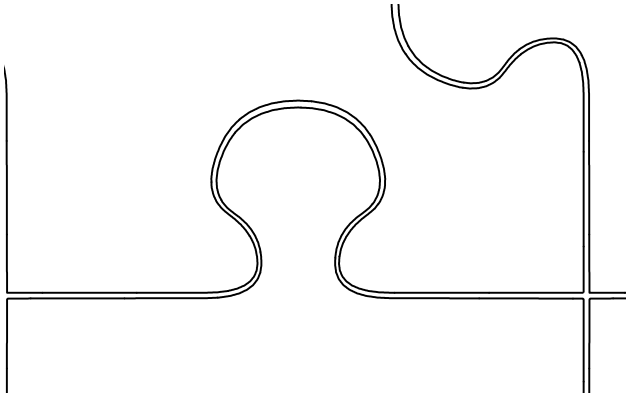

Figure 5: A zoom-in on the interior of Figure 3 (a), exposes that we offset the tiles a tad, in the domain of S, to allow for some slack once the tiles are fabricated and the puzzle is assembled.

closed curve in the domain of S. Hence, the region in 3-space of $S(T)$, can be simply created as a trimmed surface S_T, denoting S restricted to the domain inside T. This construction scheme of the 3D representation of the tiles is employed throughout this work, including in Figures 2 and 3.

If the surface $S(u, v)$ is periodic and tiles cross out of the domain of S, the surface S must be extended. Consider a closed/periodic (in u) surface $S(u, v)$, $(u, v \in [0, 1])$. Let surface $\hat{S}_u$ be the extended-in-u surface of S:

$$\hat{S}_u(u, v) = \begin{cases} S(u+1, v), & u \in [-1, 0], \\ S(u, v), & u \in [0, 1], \\ S(u-1, v), & u \in [1, 2]. \end{cases} \quad (1)$$

A similar extension can be applied in v. But now we can compose $T(t)$ even if its interlocking handles exceed beyond the domain of S, making sure all created tiles are properly constructed.

3.1.4 Adding Realizable Thickness to the Tile Shape

So far the created tiles have zero thickness. Another common operation in geometric modeling systems is making a shell model out of a single surface S. The surface S is offsetted (along the normal field of S) by the desired thickness amount, into another surface S_o, only to skin the sides of the created shell model by adding the side faces between S and S_o.

Of special care are the cases where the tiles are crossing a C^1 discontinuity in the geometry. The offset can be computed as a rounded offset around this C^1 discontinuity or a miter joint, etc. For example, moving from the base of the vase in Figure 2 (b) to the rounded lateral side of the vase, traverses a C^1 discontinuity. Figure 6 shows one such discontinuous tile, with a mitter joint along the C^1 discontinuity, from the constructed vase puzzle, in Figure 3 (b).

Figure 6: Two views of a puzzle tile that crosses a C^1 discontinuity, in the surface S. This tile is from the bottom of the vase surface in Figure 3 (b).

Remark: There are many well-known algorithms for computing the offsets of freeform surfaces [11, 21, 24, 25]. These techniques essentially deal with the trimming of offset self-intersections that may occur when the offset distance is relatively larger than the local radius of curvature or twice the global separation distance between two different parts of the surface. In the current work, the thickness of tiles is usually too small to cause these problems.

3.1.5 Randomizing the Shapes of the Puzzle Tiles

Reexamining Figure 3 (b). Because the vase is a surface of revolution, all tiles in the same height, are similar to each other. Practically, this might be undesired as we typically like individual tiles to fit the 3D puzzle in only one place. Toward this end, we propose adding an intermediate step to randomly deform all tiles a bit so each tile will end up with a unique shape.

Consider a map $M : [0, 1]^2 \rightarrow [0, 1]^2$, that is assumed to be regular. M can perturb the tiles as shown in Figure 3 (a) in a random way, and in essence, making every tile geometrically different. Since M is assumed to be regular, no singularities and certainly no self-intersections will result in the deformed tiles, due to M.

Let R_M be a B-spline surface, with open end conditions, implementing M. $R_M(u, v)$ is initialized as an identity. That is, $R_M(u, v) = (u, v)$. By randomly perturbing the interior control points of $R_M(u, v)$, we can implement this intermediate randomization composition steps, that will convert all the identical tiles in the domain of S, to geometrically somewhat different tiles. Stated differently, tile $T(t)$ will be mapped by R_M to a somewhat different

tile $T_{R_M}(t) = R_M(T(t))$. Because R_M is open-ended and we only perturbed interior control points, the boundaries will not be modified and hence preserved. The new set of deformed mapped tiles $\{T_{R_M}(t)\}$ will then be composed again, but this time with the original surface S, as before. See Figure 7, for one example, and compare with Figure 3.

3.2 A Jigsaw Puzzle of a B-rep Model Consisting of Several Trimmed Surfaces

Extending the 3D puzzle construction scheme to trimmed surfaces and solid B-rep models consisting of several trimmed surfaces, requires several steps. Before that, recall that in a B-rep model, the trimmed surfaces are stitched to their neighbors, along the trimming curves. Being a 2-manifold, the B-rep model has exactly two surfaces joining in along a trimming curve, and this adjacency information is part of the model's topology.

As a first step, the puzzle tiles in each surface must be clipped to the trimmed zone. This clipping process is discussed in Section 3.2.1. Then, clipped tiles, along a trimming curve, could possibly be stitched to other tile(s) on the other side of the trimming curve (on the adjacent trimmed surface). This optional stitching step is presented in Section 3.2.2.

3.2.1 Clipping the Tiles to the Trimming Zone

Given a set of outline curves that tile the domain of surface S, in full, as in Figure 3 (a), the problem of clipping this set of tiles to the trimmed zone, can be reduced to curve-curve intersection and Boolean operations between closed curves. Herein, we need to compute the Boolean curve intersection, in the plane (the domain of surface S), of the trimming curves of S and the closed outline curves of the tile, for all tiles. Boolean operations, and planar Boolean operations in specific, were investigated by many in the past with known algorithms for computing these operations, e.g. [4].

One could aim to adjust the shapes of the puzzle tiles in the domain of trimmed surface S, to follow and match tiles on the adjacent trimmed surface, along the shared trimming curves. While possible in simple cases, in general, the trimming curves can assume arbitrary shape and topology and hence such an approach is a major challenge on its own.

Figure 8 shows one example of a trimmed surface, with the clipped tiles to the trimmed zone, in the domain of S and in the Euclidean space.

3.2.2 Stitching Neighboring Tiles along Trimming Curves

So far, each trimmed surface is processed independently and one might select to leave it at that. This means that each trimmed surface will define a sub-puzzle that covers that trimmed surface and then all these sub-puzzles of all the individual trimmed surfaces should be glued together, to form the final model.

An alternative will be to stitch and join adjacent trimmed puzzle tiles, on both sides of a trimming curve, into one tile. In other words, if so far, all tiles were located in a single (trimmed) surface, we will now have tiles that span regions in two adjacent surfaces.

As stated, the topology is provided as part of the model, and hence we can find with ease the adjacent trimmed puzzle tiles, given a trimming curve. The two options of stitching adjacent tiles along trimming curves, or not, are depicted in Figure 9, that explodes the intersecting joint between the spout and the body of the Utah teapot – see also Figure 15. Note that one clipped tile on one side can be stitched to more than one tile on the other side. See also Figure 10. Without the stitching along trimming curves, the different trimmed surfaces will yield disjoint puzzle parts that must be glued together. In other words, this stitching makes a single 3D puzzle of the entire model. The clipped tiles on both sides of the trimming curve are independent as the two parametrizations of the two surfaces sharing the trimming curve are unrelated. Hence, tiles on one side can overlap with one or more tiles on the other side of the trimming curve. Further, a tile can only partially overlap with a tile on the other side. The stitching we employ, of the tiles along trimming curves, is heuristic: a tile on one side will be stitched with a tile on the other, only if a minimal fraction of the arc-length of the overlapping edge is shared, in either side. This fraction value is user specified, and practically was around 30%, in all presented models in the next section.

4 Results

In this section, we present some additional results. These examples include 3D printings of the puzzle models as well as their assembly. The J55 printer of Stratasys [30] was employed throughout. We converted the presented models into polygonal OBJ files that the interface of the J55 printer can read in.

The surface model of a duck in Figure 11 (a), was used to create the puzzle shown in Figure 11 (b), with each tile uniquely colored. Figure 11 (c) shows an exploded view of (b). Figure 11 (d) shows the 3D printed model after

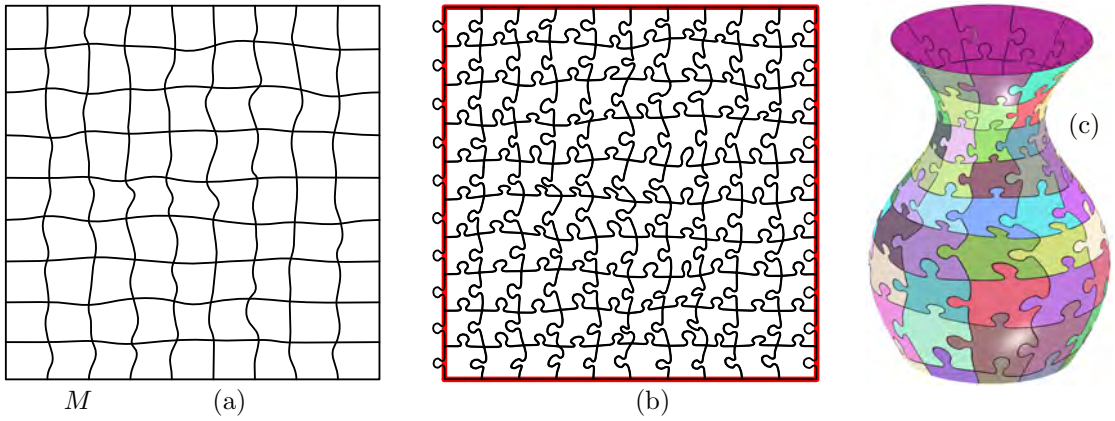

Figure 7: A randomized, yet regular, mapping $R_M : [0,1]^2 \to [0,1]^2$, in (a), is used as an intermediate composition step to deform the uniform tiles $\{T(t)\}$ from Figure 3 (a), into randomly different shapes, $\{R_M(T(t))\}$ as can be seen in (b). (c) shows the final composed result, over S, the vase from Figure 2 (b).

.

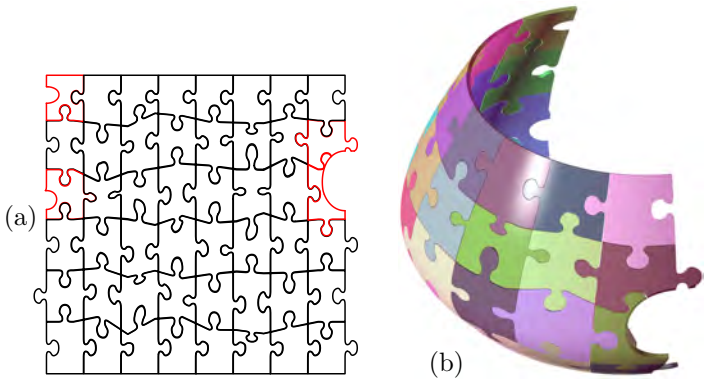

Figure 8: A full tiling of the domain of S, is clipped to the trimmed zone of the trimmed surface in (a). Clipped tiles are highlighted in red. In (b), the resulting geometry is shown mapped to 3D. Half the body of the Utah teapot (see also Figures 9 and 15).

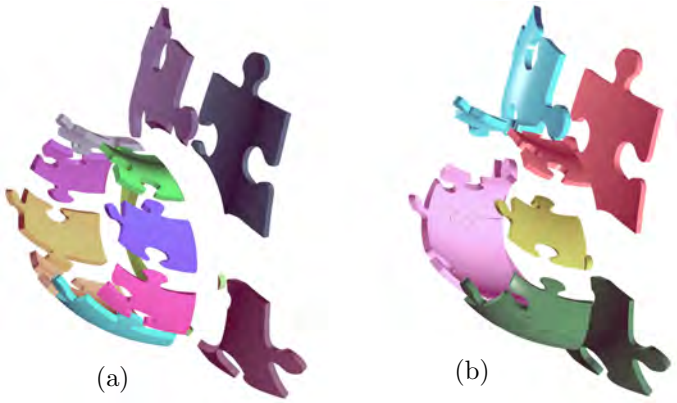

Figure 9: Across trimming curves, puzzle tiles are clipped to the trimming curve from both sides. These tiles could be left as is as shown in (a), exploded, or stitched together, as shown in (b), exploded. Different tiles are colored differently. These exploded views are from the joint between the spout and the body of the Utah teapot (see also Figures 10 and 15).

Figure 10: Tiles stitched along trimming curves in the model. These examples are from the teapot model in Figure 15.

assembling it. Finally, Figure 11 (e) shows a zoom-in view of (d), on the area of the head of the duck.

This model of the duck puzzle did not employ proper boundary conditions and at the singular head (where the Jacobian of the duck is vanishing), the interlocking handles were left unchanged, as can be seen in Figure 12 (b). Such a singular boundary, that vanishes to a point, better employs boundary tiles with no interlocking handles, as in the top of Figure 3 (a) (and also seen, for example, in the Euclidean space, in Figures 15 (g) and (h)). There are recent developments for handling the smoothness conditions at the polar singular point by constructing C^k-smooth polar splines for any $k \geq 0$ [29, 33]. These techniques may be used for resolving the computational difficulties near the polar point by eliminating the singularity in the parametric representation of freeform surface. Nevertheless, the fabrication constraints require other alternatives. A simple solution would be to replace the polar neighborhood with a disk-like capstone piece. In this example, we leave this part open so as to give an idea how the geometric features get narrower and narrower.

The puzzle of the vase model that was used as an example through this work, is shown in full in Figure 13. Figure 13 (a) shows the 3D printed and assembled tangible model, while Figurea 13 (b) and 13 (c) show two computer model views of this model, including one from below. A reparametrization R_M was employed in this model, that is a surface of revolution, to ensure all tiles are somewhat different. Going unnoticed, this model is textured with the map of the earth. Adding textures to these 3D puzzles is trivial, employing the inherent parametrization of the input surface(s) and propagating that to the individual puzzle tiles.

The model shown in Figure 14, presents an example of a solid model created using Boolean operations over freeform surfaces. It consists of four trimmed surfaces stitched together: the base sweep surface (red in Figure 14 (d)), the top surface of revolution vertical disk (green in Figure 14 (d)), the subtracted hyperbolic surface of revolution, through-hole (cyan in Figure 14 (d)), and the subtracted extruded surface in the middle bottom (yellow in Figure 14 (d)). This model is one example of a model that can benefit from randomizing the shapes of the tiles. Being symmetric in two main planes, this model yields identical puzzle parts (up to color) in different sides of the model. Whether this is a desired result or not, is up to the end user. Also, note the highly bent puzzle tiles near the top, along the highly curved edge of the surface of the input model.

Our final example is the Utah teapot that is another trimmed model. Three tensor product surfaces exist in this model – the handle, the spout and the body (that includes a bottom), three surfaces that are unioned together to form the teapot. See Figure 15. In Figure 15 (a) and (b), two computer views are shown of this puzzle. A bilinear reparametrization function R_M has been used over the body surface as it is (almost) a surface of revolution. This bilinear reparametrization yields tiles that are sometimes C^1 discontinuous. See, for example, the top edge of the marked-with-arrow tile in Figure 15 (a). While the outline curves of the tiles are C^1 discontinuous, the surfaces of the tiles assume the smoothness of the surface of the model, here, he smoothness of the Body of the Utah teapot. In other words, if the model is smooth, the tiles will remain smooth as well. Figure 15 (c) shows all the parts of this puzzle before the assembly and (d) shows a partially assembled result. Finally, Figure 15 (e) and (h) show the final assembled teapot, the first two in similar views to (a) and (b). The top edges of the body and the spout are open and terminate with tiles that have no interlocking handles, as in Figure 3. Further, the bottom of the body surface is singular at the center, as the entire bottom edge of the domain of the body is mapped to the center point as is shown in Figure 15 (g) and (h). Reexamine Figure 3 (a). The entire bottom edge collapses here to a point, while all tiles around the singularity reduce to a triangular topology.

Figure	**Computation time**	**# of tiles**
12	0.8 sec.	80
13	0.7 sec.	80
14	20.2 sec.	113
15	3.6 sec.	139
16	1.0 sec.	333

Table 1: Some statistics on the presented examples.

Table 1 provides some statistics on all the examples presented in this work. All these puzzles were synthesized (excluding 3D printing and assembly times!) in a few seconds, on a relatively modern PC running Windows 10, and using Intel Core i7-7700K 4.2 GHz CPU and 32GB of memory, in a single thread. As can be seen, most of the computation time has been devoted to handling the trimmed geometry, and specifically, the stitching of adjacent tiles along trimming curves (Figures 14 and 15).

We conclude this section with a few remarks on the employed assembly process. All the 3D printed puzzle parts, as can be seen, for example, in Figure 15(c), are 2-3 mm thick, with the motivation of reducing 3D printing costs, while still having strong enough puzzle pieces. In many cases, and in order to strengthen the assembly of the mod-

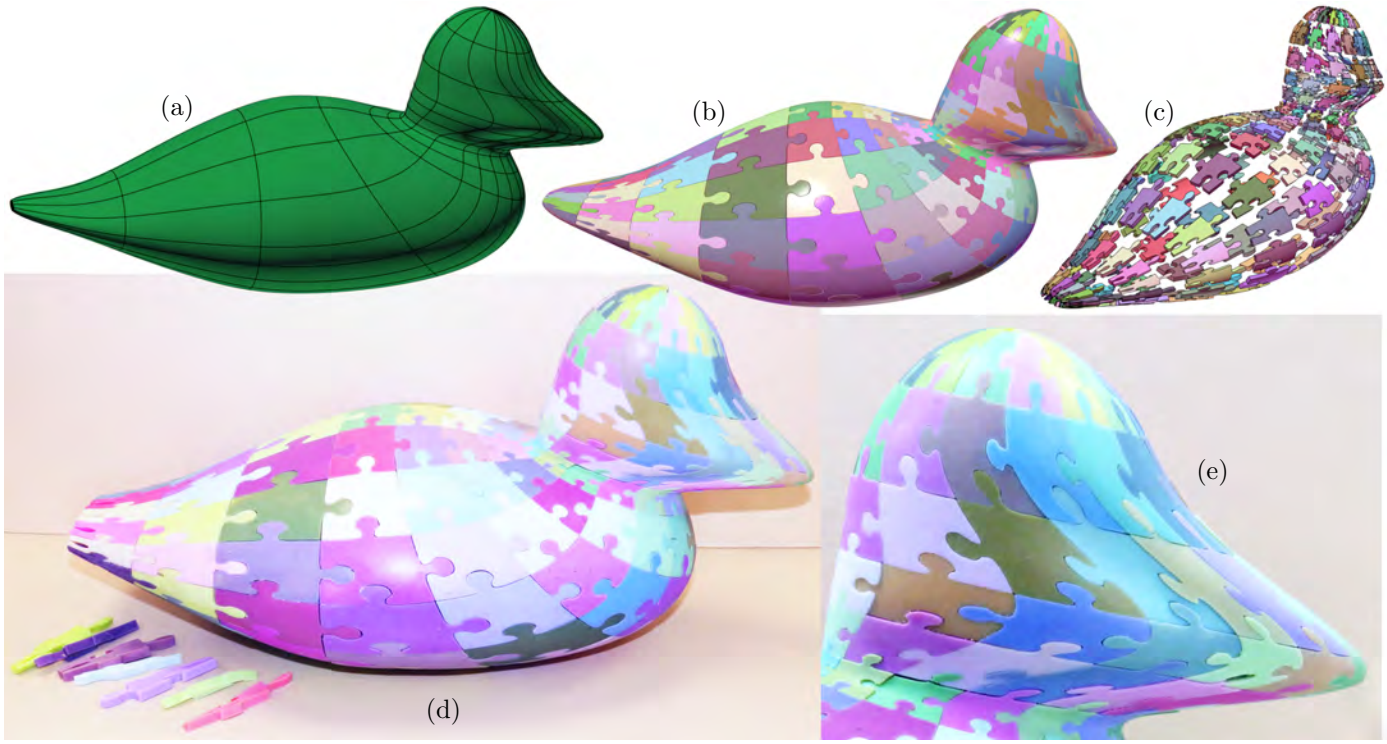

Figure 11: A duck Puzzle. The surface model of a duck in (a), is employed to create the 3D puzzle in (b), that is also shown exploded in (c). Each tile is uniquely colored. In (d), a 3D printed model of the puzzle is presented and (e) shows a zoom-in on the head. Note the tail of the duck is not fully assembled and some of the puzzle tiles near the tail are shown on the side in (d).

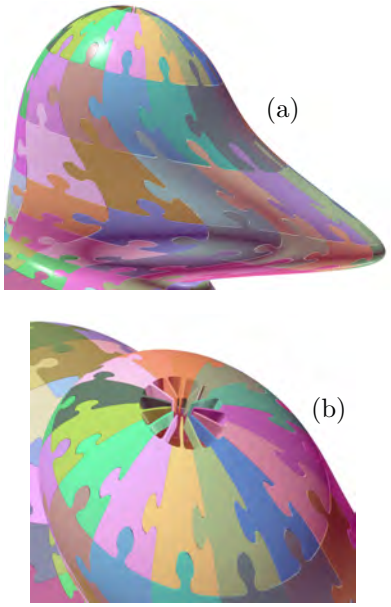

Figure 12: The head of the puzzle of the duck (a), from Figure 11, is not using proper boundary condition and hence the interlocking handles, at the tip of the head, are not closing the surface properly, as seen in (b).

els that was not very stable due to these thin walls, superglue was employed between the puzzle parts. Further, the parts that are 3D printed by the J55 printer are rigid and brittle. A careful examination of the different assembled model, will reveal some broken (and glued) small parts.

5 Conclusions and Future Work

In this work, we have presented algorithms to construct 3D jigsaw puzzles for parametric surfaces and B-rep models, possibly consisting of several trimmed surfaces.

The presented scheme can be applied to build 3D jigsaw puzzles for polygonal meshes as well. Following the presented paradigm, the polygonal mesh must support the following operations:

- The polygonal mesh must possess a parametrization P from the plane.
- Individual puzzle tiles must be clipped to the domain of P.
- The polygonal mesh should support curve-mesh composition. That is, the outline curves of the puzzle tiles should be mapped from the domain of the parametrization to 3-space, using P. This computation could be approximated using point based evaluations.
- Offset mesh operation (to give the tile some thickness) should be supported as well.

We reemphasize that tiles can be of arbitrary shape. Any covering of the plane can be used as an initial tiling arrangement for a 3D puzzle, as in, for example [19]. Figure 16 (a) shows one such example employing an M.C. Escher-style tiling of the domain of the vase surface, that is shown in (b) in 3D, and exploded in (c). Note the alternative way we ensure periodicity here, having the pieces of the lizard tiles clipped, while they perfectly fit the domain of the surface, along the closed seam, in (b). This approach can be employed when a closed, yet non-perodic, B-spline surface is provided, in which case all tiling geometry must not exceeed the surface domain, for proper functional composition.

While any covering of the plane can indeed be employed, practically, tiles that are very small were difficult to assemble and even broke at times. Hence, the designer of the puzzle should take that into consideration, possibly by exploiting and controlling the parametrization of the surfaces, avoiding too-small tiles, especially near trimming curves.

The J55 printer of Stratasys was used here to 3D print the tiles. Accuracy plays a crucial part in fabricating a viable puzzle, and here the J55 performed superbly. However, the 3D printed material is rigid and brittle as already stated. Commercial 3D puzzles are commonly created from a soft material that is flexible (e.g., foam), and hence more suitable to assemblies and puzzles. Further, foam pieces are typically thicker (close to 7 mm) which makes them better at self-supporting the structure without any glue. On the other hand, some commercial puzzles (as shown in Figure 1) are made of sturdy plastics and thus thinner (close to 2 mm). (The shoe puzzle even has a small number of doubly-curved tiles.) Such concerns should be taken into considerations, in practical 3D puzzles.

In this work, we randomized the tile while keeping the boundary intact. Clearly, and using the surface extension ability discussed in Section 3.1.3, the randomization of R_M, as discussed in Section 3.1.5 can also be extended to operate over periodic boundaries. More generally, and as already stated, the inherent regular parameterization of a surface can have a significant impact on the shapes and sizes of the resulting puzzle tiles, over that surface. This is a degree of freedom that clearly depends on the shape of the surface, and the stretch properties of the map of the surface, or its first fundamental form [7], that the designer of the puzzle should be aware of, and possible exploit.

Finally and as stated, the stitching process, of tiles along trimming curves, is heuristic. Alternative stitching heuris-

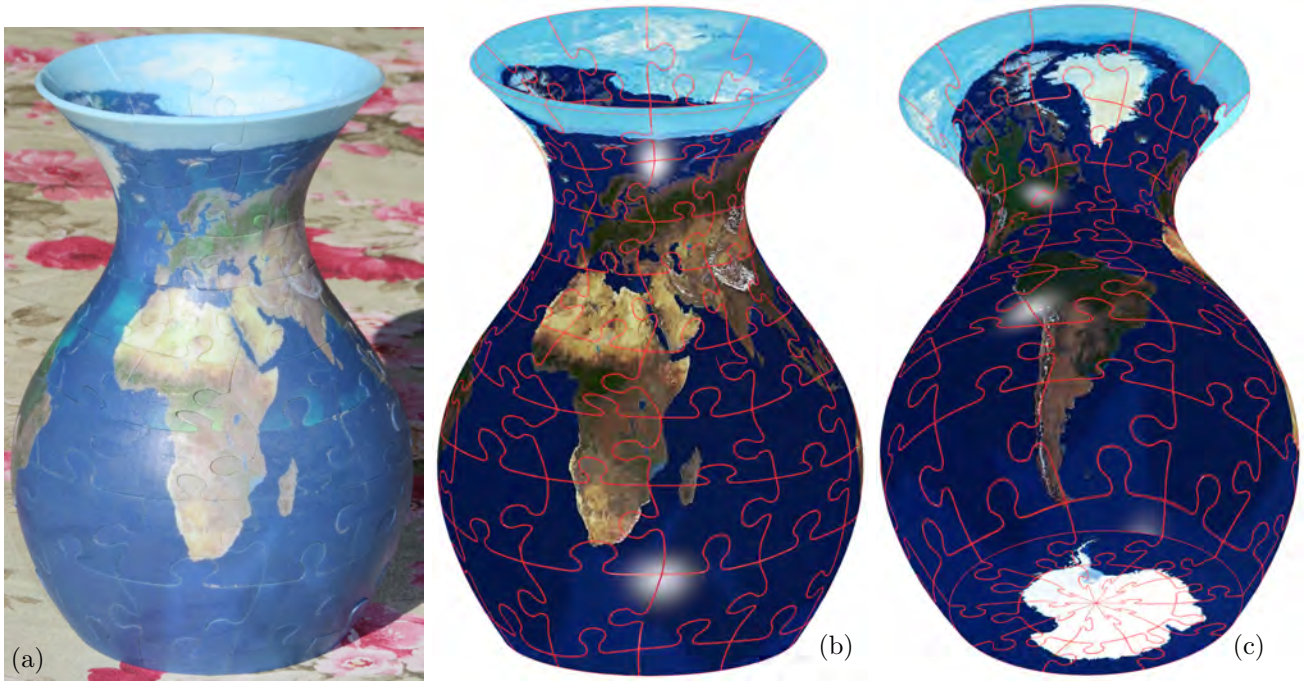

Figure 13: The vase Puzzle used throughout this work is shown in full. In (a), a 3D printed example is presented while (b) and (c) show two different views of the computer model.

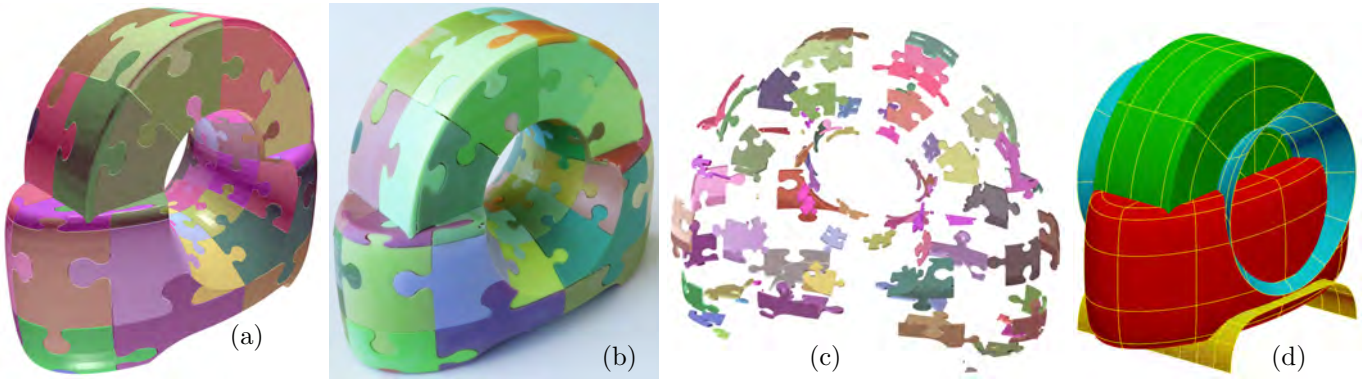

Figure 14: A model that is the result of several Boolean operations over freeform surfaces (a). The resulting 3D printed and assembled puzzle (b) has pieces that are stitched along the shared seams (trimming curves) of the surfaces, seams that are C^1 discontinuous. In (c), an exploded view of the puzzle from (a) is presented. Note that some puzzle pieces are quite tiny. (d) shows the original surfaces used to create this model.

Figure 15: A puzzle of the Utah Teapot, that is a model consisting of three trimmed surfaces. (a) and (b) show the computer models, whereas (c) shows the 3D printed parts and (d) the partially assembled model. Finally, in (e) to (h), views of the assembled teapot are shown, the first two in similar views to (a) and (b), while the last two shows the bottom of the teapot.

Figure 16: M. C. Escher style tiling in (a) that is employed to create the 3D jigsaw puzzle that is shown in (b) and exploded in (c). Note in (b), the alternative way we ensure periodicity here with lizard tile pieces that perfectly align along the closed seam.

tics might be helpful and more intuitive.

All the C source code of the implementations of the Jigsaw algorithms, as presented in this work, is available as part of the IRIT solid modeling system [13]. Also provided in [13] are scripts to recreate the Jigsaw examples presented in this work.

Acknowledgments

We would like to thank anonymous reviewers for their invaluable comments. This research was supported in part by the European Union Horizon 2020 research and innovation programme, under grant agreement No 862025, in part by the ISRAEL SCIENCE FOUNDATION (grant No. 597/18), and in part by the National Research Foundation of Korea (No. NRF-2019R1A2C1003490, NRF-2019K1A3A1A78112596).

References

[1] A. Bermano, T. Funkhouser, S. Rusinkiewicz: State of the art in methods and representations for fabrication-aware design. *Computer Graphics Forum*, **36**(2):509-535, 2017.

[2] B. Bickel, P. Cignoni, L. Malomo, N. Pietroni: State of the art on stylized fabrication. *Computer Graphics Forum*, **37**(6):325-342, 2018.

[3] D. Bommes, B. Lévy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, D. Zorin: Quad-mesh generation and processing: A survey. *Computer Graphics Forum*, **32**(6):51-76, 2013.

[4] CGAL: *2D Regularized Boolean Set-Operations*. https://doc.cgal.org/latest/Boolean_set_operations_2/index.html

[5] S. Coffin: *Geometric Puzzle Design*. A.K. Peters, Wellesley, MA, 2006.

[6] E. Cohen, R.F. Riesenfeld, G. Elber: *Geometric Modeling with Splines: An Introduction*, AK Peters, Wellesley, MA, 2001.

[7] M. P. DoCarmo. Differential Geometry of Curves and Surfaces. Prentice-Hall, 1976.

[8] G, Elber and M.-S. Kim: Euclidean offset and bisector approximations of curves over freeform surfaces. *Computer Aided Geometric Design*, **80**, Article 101850, 2020.

[9] G. Farin: *Curves and Surfaces for CAGD: A Practical Guide*, 5th Ed., Morgan Kaufmann, San Francisco, CA, 2002.

[10] S. Golomb: *Polyominoes: Puzzles, Patterns, Problems, and Packings*, Revised second edition. Princeton University Press, Princeton, NJ, 1994.

[11] Q Hong, Y. Park, M.-S. Kim, G. Elber: Trimming offset surface self-intersections around near-singular regions. *Computers & Graphics*, **82**:84-94, 2019.

[12] J. Hoschek and D. Lasser: *Fundamentals of Computer Aided Geometric Design*. AK Peters, Wellesley, MA, 1993.

[13] IRIT 12.0 User's Manual. The Technion—IIT, Haifa, Israel, 2021. Available at `http://www.cs.technion.ac.il/~irit` .

[14] C.S. Kaplan: Semiregular patterns on surfaces. *NPAR '09: Proc. of the 7th Int'l Symp. on Non-Photorealistic Animation and Rendering*, pp. 35—39, August 2009.

[15] C.S. Kaplan and D.H. Salesin: Escherization. *Proc. of ACM SIGGRAPH 2000*, pp. 499—510.

[16] N. Kita and T. Saito: Computational design of generalized centrifugal puzzles. *Comput. Graph.*, **90**:21-28, 2020.

[17] N. Kita and K. Miyata: Computational design of polyomino puzzles. *The Visual Computer*. **37**:777-787, 2021.

[18] C. Lau, Y. Schwartzburg, A. Shaji, Z. Sadeghipoor, S. Süsstrunk: Creating personalized jigsaw puzzles. *Proc. of the ACM Workshop on Non-photorealistic Animation and Rendering*, pp. 31-39. New York, NY, 2014.

[19] X. Liu, L. Lu, A. Sharf, X. Yan, D. Lischinski, C. Tu: Fabricable dihedral Escher tessellations. *Computer-Aided Design*, **127**, Article 102853, 2020.

[20] K.Y. Lo, C.W. Fu, H. Li: 3D polyomino puzzle. *ACM Trans. on Graphics*, **28**(5), 157:1-157:8, 2009.

[21] T. Maekawa, W. Cho, N.M. Patrikalakis: Computation of self-intersections of offsets of Bézier surface patches. *Journal of Mechanical Design: ASME Transactions*, **119**(2):275-283, 1997.

[22] F. Massarwi, B. van Sosin, G. Elber: Untrimming: Precise conversion of trimmed-surfaces to tensor-product surfaces. *Computers & Graphics*, **70**:80-91, 2018.

[23] N.M. Patrikalakis and L. Bardis: Offsets of curves on rational B-spline surfaces. *Engineering with Computers*, **5**(1):39-46, 1989.

[24] N.M. Patrikalakis and T. Maekawa: *Shape Interrogation for Computer Aided Design and Manufacturing*, Springer-Verlag, Heidelberg, 2002.

[25] J.-K. Seong, G. Elber, M.-S. Kim: Trimming local and global self-Intersections in offset curves/surfaces using distance maps. *Computer-Aided Design*, **38**(3):183-193, 2006.

[26] C.H. Séquin: Prototyping dissection puzzles with layered manufacturing. *Fabrication and Sculpture Track, Shape Modeling International 2012*.

[27] P. Song, C.W. Fu, D. Cohen-Or: Recursive interlocking puzzles. *ACM Trans. on Graphics*, **31**(6), 128:1-128:10, 2012.

[28] P. Song, Z. Fu, L. Liu, C.W. Fu: Printing 3D objects with interlocking parts. *Computer Aided Geometric Design*, **35**:137-148, 2015.

[29] H. Speleers and D. Toshniwal: A general class of smooth rational splines: application to construction of exact ellipses and ellipsoids. *Computer-Aided Design*, **132**, Article 102982, 2021.

[30] Stratasys: *The J55 printer*. `https://www.stratasys.com/3d-printers/-j55`

[31] T. Sun and C. Zheng: Computational design of twisty joints and puzzles. *ACM Trans. on Graphics*, **34**(4), 101:1-101:11, 2015.

[32] K. Tang, P. Song, X. Wang, B. Deng, C.W. Fu, L. Liu: Computational design of steady 3D dissection puzzles. *Computer Graphics Forum*, **38**(2):291-303, 2019.

[33] D. Toshniwal, H. Speleers, R. Hiemstra, T. Hughes: Multi-degree smooth polar splines: A framework for geometric modeling and isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, **316**:1005-1061, 2017.

[34] F.-E. Wolter and S. Tuohy. Approximation of high-degree and procedural curves. *Engineering with Computers*, **8**(2):61-80, 1992.

[35] S. Xin, C.F. Lai, C.W, Fu, T.T. Wong, Y. He, D. Cohen-Or: Making burr puzzles from 3D models. *ACM Trans. on Graphics*, **30**(4), 97:1-97:8, 2011.

[36] Y. Yuan, C. Zheng, S. Coros: Computational design of transformables. *Computer Graphics Forum*, **37**(8):103-113, 2018.

[37] Y. Zhou, S. Sueda, W. Matusik, A. Shamir: Boxelization: folding 3D objects into boxes. *ACM Trans. on Graphics*, **33**(4), 71:1-71:8, 2014.