Fabricatable 90° Pop-ups: Interactive Transformation of a 3D Model into a Pop-up Structure

J. Fujikawa[1] and T. Ijiri[1]
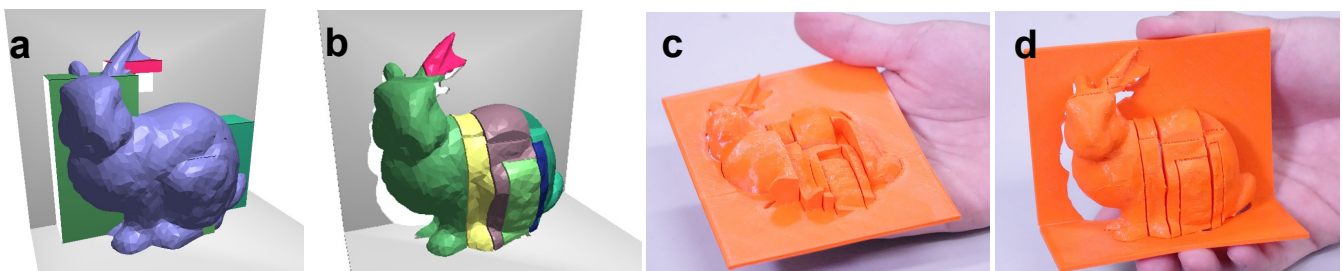
[1]Shibaura Institute of Technology, Japan

Figure 1: *A bunny model represented with a fabricatable 90° pop-up. The user designs a planar pop-up that roughly approximates the input 3D model (a). The proposed method automatically deforms all pop-up components while preserving their fabricatability (b). The pop-up is printed in the unfolded state (c) and provides a 3D shape when folded 90° (d).*

Abstract
Ninety-degree pop-ups are a type of papercraft on which a three-dimensional (3D) structure pops up when the angle of the base fold is 90°. They are fabricated by cutting and creasing a single sheet of paper. Traditional 90° pop-ups are limited to 3D shapes only comprising planar shapes because they are made of paper. In this paper, we present novel pop-ups, fabricatable 90° pop-ups that employ the 90° pop-up mechanism, consist of components with curved shapes, and can be fabricatable using a 3D printer. We propose a method for converting a 3D model into a fabricatable 90° pop-up. The user first interactively designs a layout of pop-up components, and the system automatically deforms the components using the 3D model. Because the generated pop-ups contain necessary cuts and folds, no additional assembly process is required. To demonstrate the feasibility of the proposed method, we designed and fabricated various 90° pop-ups using a 3D printer.

CCS Concepts
• Computing methodologies → Graphics systems and interfaces; Shape modeling; • Applied computing → Computer-aided manufacturing;

1. Introduction

Pop-ups are a type of papercraft on which three-dimensional (3D) structures pop up when the base paper is unfolded. They can be used as greeting cards and books. They also have industrial applications, such as converting micro two-dimensional (2D) patterns into 3D structures [HHR00] and analyzing the folded structure of proteins [SA01].

The 90° pop-ups, also called origamic architectures [Cha85], are a class of pop-ups in which a 3D shape appears when the base paper is folded 90°. They can be created by cutting and creasing a sheet of paper. In the graphics community, various methods of converting 3D models into 90° pop-ups have been developed [LSH*10, LLLN*14]. However, because traditional 90° pop-ups are composed of paper, they can only produce a combination of flat planes as well as are difficult to represent 3D models with curved surfaces.

Our goal is to represent various 3D models based on the 90° pop-up mechanism. We propose a novel pop-up representation, *fabricatable 90° pop-ups*, which employs the 90° pop-up mechanism, comprises components with curved shapes, and can be fabricated using a 3D printer. We also propose an interactive method for converting a 3D model into a fabricatable 90° pop-up. The user designs

a planar 90° pop-up that roughly approximates the input 3D model. Subsequently, each planar component is deformed so that the entire pop-up better approximates the input model while preserving its fabricatability.

We demonstrated the feasibility of the proposed method by designing and fabricating various 90° pop-ups. Using a 3D printer, we fabricated 90° pop-ups that represent 3D models with curved surfaces. Among them is a 90° pop-up bunny, as shown in Figure 1. The proposed method allows the user to design a fundamental pop-up structure in less than 10 min of interaction. As the obtained pop-ups already contain the necessary cuts and folds, no additional assembly process is required. Furthermore, by printing the mechanisms in their unfolded state, we can reduce most of the support materials.

The main contributions of this paper are summarized as follows:

1. We propose fabricatable 90° pop-ups that can approximate various 3D models with a pop-up mechanism and can be fabricated using a 3D printer without requiring an assembly process.
2. We present a method for interactively designing pop-up structures to convert 3D models into fabricatable 90° pop-ups.
3. We modeled and fabricated various 90° pop-ups using the proposed method to demonstrate its feasibility.

The software and source code of our prototype system are available at https://github.com/InteractiveGraphicsLab/Fabricatable-90-Pop-up.

2. Related Work

2.1. Fabrication with 3D Printer

In the graphics community, several techniques for fabricating objects with various functions have been developed. For instance, methods of designing objects that self-stand [PWLSH13], spin [BBO*10], float [KUS*21], and fly [UKSI14] have been reported. Transformable objects have also been designed by integrating various mechanisms, such as gears [ZAC*17] or linkages [CTN*13]. Furthermore, deformable objects have been developed using flexible materials [YP21].

Four-dimensional (4D) printing, a technique to fabricate objects automatically deforming in response to environmental stimuli, such as heat, has been studied. An et al. [ATG*18] generated a self-folding sheet by printing multiple materials with different thermal shrinkage properties in layers. Deng et al. [DC15] and Narumi et al. [NKS*23] created self-folding objects by placing constraint layers fabricated with photocurable resin on a heat shrink sheet. Also, Kwok et al. [KWD*15] and Jian et al. [JDZ*22] proposed methods to convert a 3D model into a 4D printable form by flattening a 3D model into a planar shape. However, the pop-up mechanism has not been integrated in the previous studies discussed above.

2.2. Computational Papercraft

Papercraft is an important topic in computer graphics. Many studies have investigated the analysis and creation of papercrafts using computers. Origami is a traditional paper folding art, and its folding algorithms have been widely studied [Hul94]. Tachi proposed an algorithm that converts polyhedra into origami structures [Tac10]. Mitani developed a technique for reproducing rotational sweep shapes with origami [Mit09]. Papercutting is another type of papercraft that represents 2D pictures by cutting a sheet of paper. Xu et al. developed a tool for designing valid papercutting patterns [XKM07].

Instead of starting with a sheet of paper, a 3D shape can be generated by gluing multiple paper patches together [MS04b, STL06, ZFO*22]. The basic idea is to segment a 3D model into locally developable patches and glue them to create papercraft toys. Various 3D shapes have been achieved through this technique. However, since the paper has a locally planar shape, it is difficult for these methods to represent curved freeform shapes.

2.3. Computational Pop-ups

Pop-ups can be classified into two groups: 90° and 180° pop-ups. A 90° pop-up provides 3D shapes when the base paper is folded by 90°. It can be fabricated by cutting and creasing a sheet of paper. In contrast, 180° pop-ups show 3D shapes when the base papers are unfolded by 180°, and they can be fabricated by gluing multiple components on the base paper. Computer-aided design methods have been explored for both types of pop-ups.

Mitani and Suzuki proposed a system for interactively designing 90° pop-ups [MS04a]. The system also examines whether the user-designed pop-ups work well by simulating their motions. Various methods have been developed to automatically generate 90° pop-ups from 3D models. Li et al. proposed an algorithm for automatically converting 3D models to 90° pop-ups using a voxelization technique [LSH*10]. Le et al. proposed an algorithm that uses a 3D model and its rendering image to determine the placement of components [LLLN*14]. These two papers also discussed the constraints a valid pop-up mechanism should satisfy, i.e., foldability and stability. The constraints of our fabricatable 90° pop-ups (Sec. 3.3) are designed inspired from them.

Various methods have also been developed for modeling 180° pop-ups. Because 180° pop-ups are fabricated by gluing multiple components on the base papers, they allow more varieties of components than 90° pop-ups. Glassner analyzed four 180° pop-up mechanisms and defined their mathematical models [Gla02a, Gla02b]. Okamura et al. [OI09] and Iizuka et al. [IEM*11] proposed interactive systems that allow users to design and simulate 180° pop-ups. Li et al. [LJGH11] and Ruiz et al. [JLYL14] proposed methods for converting 3D models into 180° pop-ups.

Various pop-up models have been achieved, interactively or automatically, using these methods. However, it is still difficult to represent 3D shapes with curved surfaces since they use flat paper. To address this challenge, we propose adding detailed shapes to each component by fabricating 90° pop-ups using a 3D printer.

3. Fabricatable 90° Pop-up Mechanisms

This session introduces the definitions of fabricatable 90° pop-up mechanisms (Sec. 3.1). We then describe 3D printing parameters (Sec. 3.2) and the constraints that the fabricatable 90° pop-ups must satisfy (Sec. 3.3).

3.1. Fabricatable 90° Pop-ups and Components

Traditional 90° pop-ups are fabricated by cutting and creasing a single sheet of paper. They have entirely flat shapes when the base paper is unfolded 180° and show 3D shapes when the base paper is folded 90°. The detailed formulation has been reported in previous studies [LLLN*14, LSH*10]. Similar to previous studies, we focus on parallel pop-ups, in which all folds are parallel.

To simplify the interactive design process, we introduce a component-based representation for 90° pop-ups. In our representation, a pop-up consists of multiple components placed on a base component in a hierarchical arrangement (Figure 2). The base component that the user can fold and unfold consists of two base patches sharing a base fold. We call the angle of the base fold the fold angle $\theta \in [90°, 180°]$. The pop-up component consists of horizontal and vertical patches sharing a convex fold. The vertical and horizontal patches are linked to another patch by concave folds. This component-based representation is considered a subset of the general representation [LLLN*14, LSH*10].

We represent pop-ups consisting of multiple components using a tree data structure, where the base component is the root node, and the pop-up component bridging over the concave fold is a child of the component that possesses the concave fold. Figure 3 shows an example of a tree data structure.
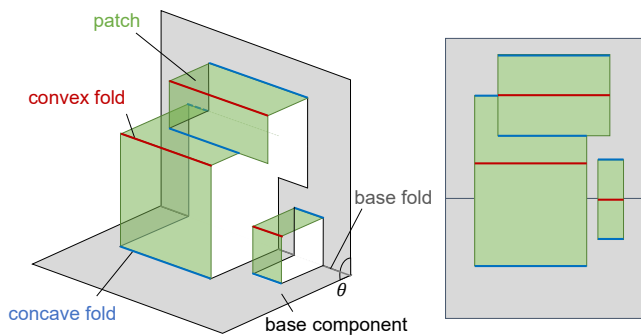

Figure 2: *Example of a 90° pop-up. Multiple components (green) are placed on the base component (gray), hierarchically. The concave, convex and base folds are highlighted in blue, red and gray, respectively.*
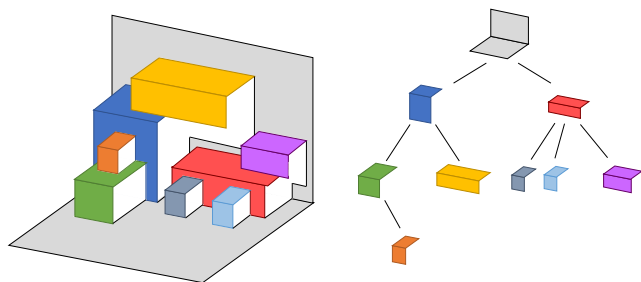

Figure 3: *A tree data structure representing the hierarchical components.*

To fabricate 90° pop-ups using a 3D printer, we extend the traditional 90° pop-up such that all components have thickness and
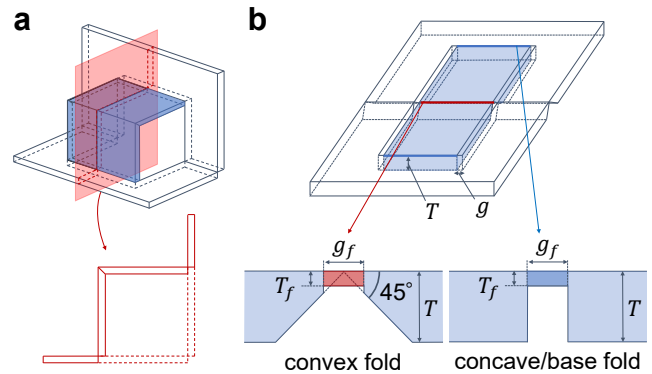

Figure 4: *An illustration of fabricatable pop-up in the 90° folded state (a) and 180° unfolded state (b).*

all folds have thin surface structures. Figure 4 shows the proposed fabricatable 90° pop-up. Each patch has the same thickness T. At each of the concave and convex folds, we place a gap g_f between two patches and connect them with a thin surface of thickness T_f at the top. We also make a 45° bevel on the convex fold such that it can fold 90° and provide a gap of thickness g between disconnected patches so that the margin is maintained during printing. The different designs are applied for the concave and convex folds since they are folded in opposite orientations.

We fabricate pop-ups in an unfolded state ($\theta = 180°$) using a fused deposition modeling (FDM) printer, which allows a significant reduction of the support material. Also, we use flexible materials, thermoplastic polyurethane (TPU), so that the concave and convex folds can bend. Because we use a 3D printer instead of paper, we can modify the shape of each patch. Furthermore, if we limit the modification to the space defined by segmenting the entire space using patch-swept cuboids, no interference occurs when the fold angle changes. We modify each patch within this space to generate various pop-up shapes (Sec. 5).

3.2. Printing Parameters

We suppose to print pop-ups with TPU using an FDM printer. Our pop-up mechanism has four parameters: the patch thickness T, fold thickness T_f, patch gap g, and fold gap g_f. We should determine these parameters such that each patch is thin but has sufficient stiffness, each gap is also thin but does not stick together during printing, and each fold is easy to fold.

To determine these parameters, we conducted a simple experiment using an FDM 3D printer, QIDI TECH's X-pro with a 0.4 mm nozzle, and three TPU materials, eSUN eTPU-95A, Sain Smart TPU and Overture TPU 3D Printer Filament. The fold thickness should be as small as possible; thus, we set it to the default laminate pitch of the printer $T_f = 0.2$ mm. We fabricated multiple pairs of two patches connected with a single concave fold by varying the patch thickness and fold gap. We varied the patch thickness from 1.0 to 3.0 mm at a 1.0-mm interval and the fold gap from 0.1 to 0.7 mm at a 0.2-mm interval. Then, we observed the fabricated patches.

We obtained similar results for all materials. We report the result

with eSUN eTPU-95A (Figure 5). With the patch thickness $T = 1.0$ mm, the patches lacked stiffness and tended to deform easily when bending the fold. In contrast, with the thickness $T \geq 2.0$ mm, the patches were stiff enough to maintain their shapes. Also, with the fold gap $g \leq 0.3$ mm, we often found that the patches stuck together. On the other hand, with a larger gap $g \geq 0.5$ mm, we rarely found such sticking. Based on these results, the following parameters were employed in this study: $T = 2.0$ mm, $T_f = 0.2$ mm, $g = 0.5$ mm, and $g_f = 0.5$ mm. These parameters should be tuned depending on the printer device and materials.
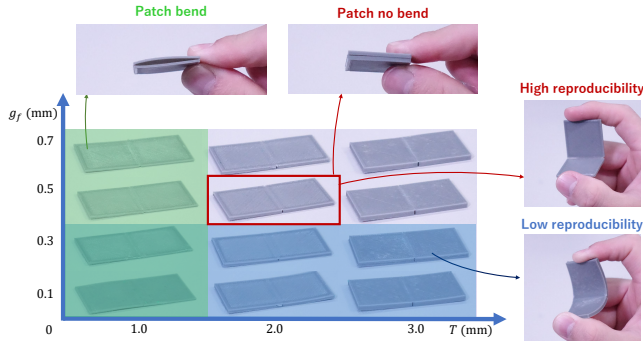

Figure 5: *Pairs of patches printed with different patch thicknesses T and fold gaps g_f. With $T \leq 1.0$ mm (green area), patches were not stiff enough. With $g_f \leq 0.3$ mm (blue area), we often found the patches stuck together.*

3.3. Topological and Geometrical Constraints

For the fabricatable 90° pop-ups to work appropriately, they must satisfy two topological constraints (connectivity and no-overlap constraints), and three geometrical constraints (maximum size, minimum height, and minimum width and fold length). Specifically, the topological constraints are necessary for ensuring that the pop-up mechanisms work well. On the other hand, the geometrical constraints arise by adding thickness to components and ensure the pop-ups fabricatable.

The connectivity constraint is for ensuring that all components move and synchronize with the base component. This constraint is satisfied when all components have their two concave folds placed on two patches of different components. It is not satisfied if there is a component whose concave fold floats or is not attached to other patches. Figure 6ab shows the components that satisfy and do not satisfy this constraint.

The no-overlap constraint is for ensuring that all components can be constructed from a single sheet. This constraint is satisfied when each component does not intersect with any other components except its parent in the unfolded state (Figure 6ac).

The maximum-size constraint limits the entire pop-up size to ensure it can be fabricated using a 3D printer. A pop-up satisfies this constraint when its bounding box in the unfolded state is smaller than the printing volume of a 3D printer.

The minimum-height constraint restricts the height of the
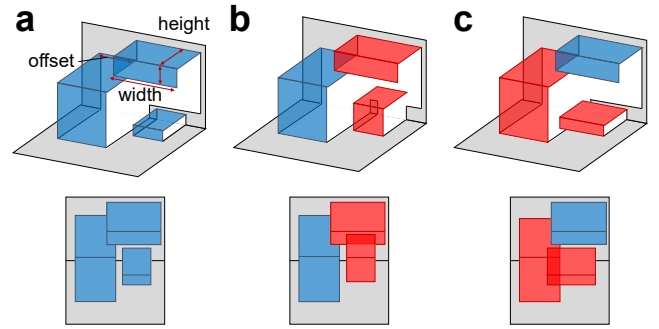

Figure 6: *An illustration of the constraints. Three components in (a) satisfy the connectivity and no-overlap constraints. Red components in (b) and (c) break connectivity and no-overlap, respectively.*

patch, where the height refers to the length of the patch perpendicular to its folds (Figure 6a). The height of each patch should be greater than the patch thickness T, which is required to add 45° bevels at the convex fold.

The minimum-width-and-fold-length constraint restricts patch width and fold length. Since a 3D printer is used to fabricate the pop-ups, these minimum values are determined by the diameter of the printer nozzle, which was 0.4 mm for the printer used in this study. While we set the minimum value to the nozzle diameter, specifying a higher value could enhance the robustness of the pop-up structure.

4. Interactive Modeling for Planer Pop-ups

The proposed 90° pop-ups can be modeled in two steps. In the first step, the user interactively creates a pop-up composed of planar components without a thickness that roughly approximates an input 3D model. In the next step, the proposed system automatically modifies the shape of all components such that the pop-up better approximates the model (Sec. 5). We develop a design tool that allows the user to place, move, modify, and simulate pop-up components.

Figure 7a shows a screenshot and interaction process of the proposed tool. The user first loads a target 3D model so that it can be referenced during modeling. The visualization of the model can be turned on and off by pressing the shift key. When the user presses the *Place* button and clicks on a concave fold (Figure 7b, red line), a new component is placed at the center of the clicked fold. This interaction was designed based on previous methods [OI09, MS04a]. When the user presses the *Resize and Translate* button and clicks on an existing component, the system presents a handle consisting of three orthogonal arrows. The user can drag each arrow to modify the height, width, and position of the component. When the user presses the *Delete* button and clicks on an existing component, the component is removed, and if the component has children, all descendant components are removed.

After the user adds or deforms components, the tool checks whether the connectivity and no-overlap constraints are satisfied. When the constraints are not satisfied, the tool provides a warning, and when the constraints are satisfied, it computes a 2D component

layout by unfolding all components and placing them sequentially from the top to the bottom of their tree structure. When placing a component, existing 2D components are cut out to make space for the new component.

The proposed tool allows the user to simulate the motion of the designed pop-up mechanism. There is a slider below the main window that the user can drag to modify the fold angle of the base component $\theta \in [90°, 180°]$ to check the motion of the designed pop-ups. During the simulation, the motion of each vertex, $(x(\theta), y(\theta), z(\theta))$, of the mechanism can be calculated based only on θ [MS04a], as follows:

$$\begin{cases} x(\theta) = x_p - y_p \cos(180° - \theta) \\ y(\theta) = y_p \sin(180° - \theta) \\ z(\theta) = z_p \end{cases} \qquad (1)$$

where (x_p, y_p, z_p) are the positions of the vertices in the folded state, $\theta = 90°$, and all vertices are placed in the coordinated system where the x-, y-, and z-axes are aligned as shown in Figure 8.

5. Automatic Transformation for Planar Pop-ups into Fabricatable Forms

The user-designed pop-up consisting of planar and rectangular patches can be transformed into a fabricatable form in four steps: thickening the patches, trimming the patch outlines, transforming the patch shapes, and adding gaps and bevels to the folds.

5.1. Thickening the Patches and Trimming Their Outlines

We consider the unfolded state of the pop-up, $\theta = 180°$, and add a thickness T to the patches of all components. The bottom surface attaches to the printing bed, and two patches linked with a fold are connected at the top surface (Figure 9b).

Next, the outlines of all patches are trimmed. The pop-up is folded, $\theta = 90°$, and the target 3D model $\mathcal{M}$ is overlaid on it. Then, the shape of each patch $\mathcal{P}_i$ is modified as follows:

$$\mathcal{P}'_i = \mathcal{P}_i \cap \mathcal{M} \qquad (2)$$

where $\cap$ is the intersection of the constructive solid geometry (CSG) operation. Notably, each patch is linked to two adjacent patches with the concave and convex folds. If each of the two folds is entirely eliminated or if a patch is separated, losing the connectivity between the two folds, the pop-up mechanism is broken. When such a patch is observed, the trimming operation is skipped, and its original rectangular shape is used. This outline trimming creates empty spaces outside the outlines in the unfolded state (Figure 9c), which can be eliminated by adjusting the patches inward, ensuring that only a gap of width g remains.

5.2. Transforming the Patches via Spatial Segmentation

The pop-ups must be able to fold and unfold, $\theta \in [90°, 180°]$, without their components interfering with each other. This requirement can be satisfied even when each component patch is deformed within a space defined by sweeping it in its normal direction and subtracting the swept spaces of other patches in the folded state.
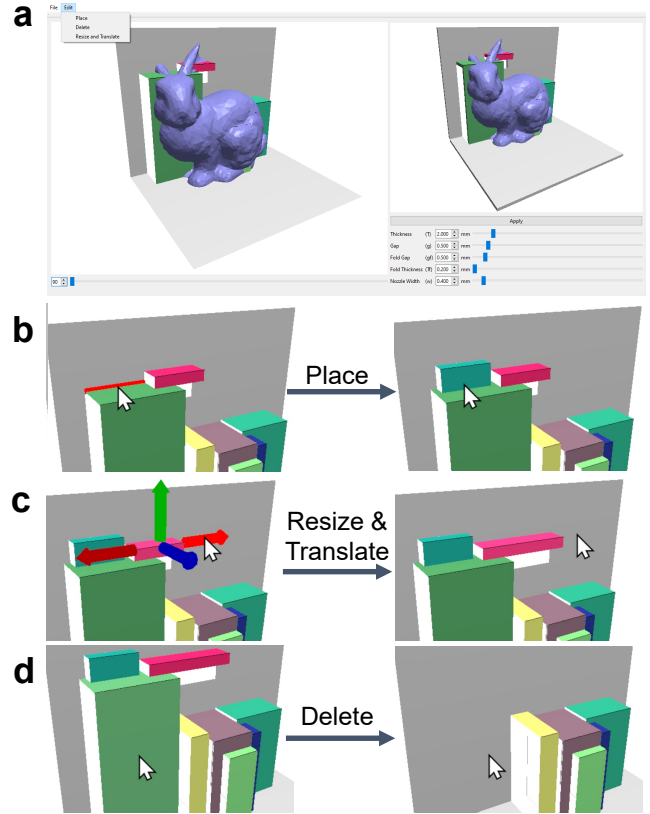
Figure 7: *User interface for creating a pop-up. The pane (a) shows a screenshot of our tool. The user adds a new component by clicking a target concave fold (b), modifies its shape by dragging the handle (c), or deletes a component by clicking it (d).*
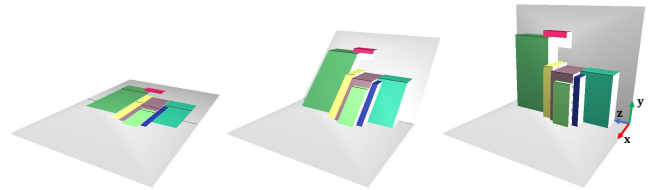

Figure 8: *By manipulating the slide bar, the user can simulate the pop-up motion.*

Based on this observation, we transform all patches such that the entire pop-up closely approximates the 3D model (Figure 9d).

To compute the shape of each patch, we segment the 3D space into regions associated with the patches. We perform this segmentation by using the pop-up with rectangular patches before trimming and considering its 90° folded state. Let us consider the coordinate system where the x-, y-, and z-axes are aligned as in Figure 10a. We define the entire space Ω by the bounding box of the base component. For each component C_i, we define four sweep regions, S_i^{+x}, S_i^{-x}, S_i^{+y}, and S_i^{-y} (Figure 10cd). S_i^{+x} and S_i^{-x} are defined by sweeping the vertical patch of C_i along the positive and negative directions of the x-axis in Ω, respectively, and S_i^{+y} and S_i^{-y} are ob-

a. User design　　**b.** Thickening patches　　**c.** Trimming outlines　　**d.** Transforming patches　　**e.** Printing pop-up
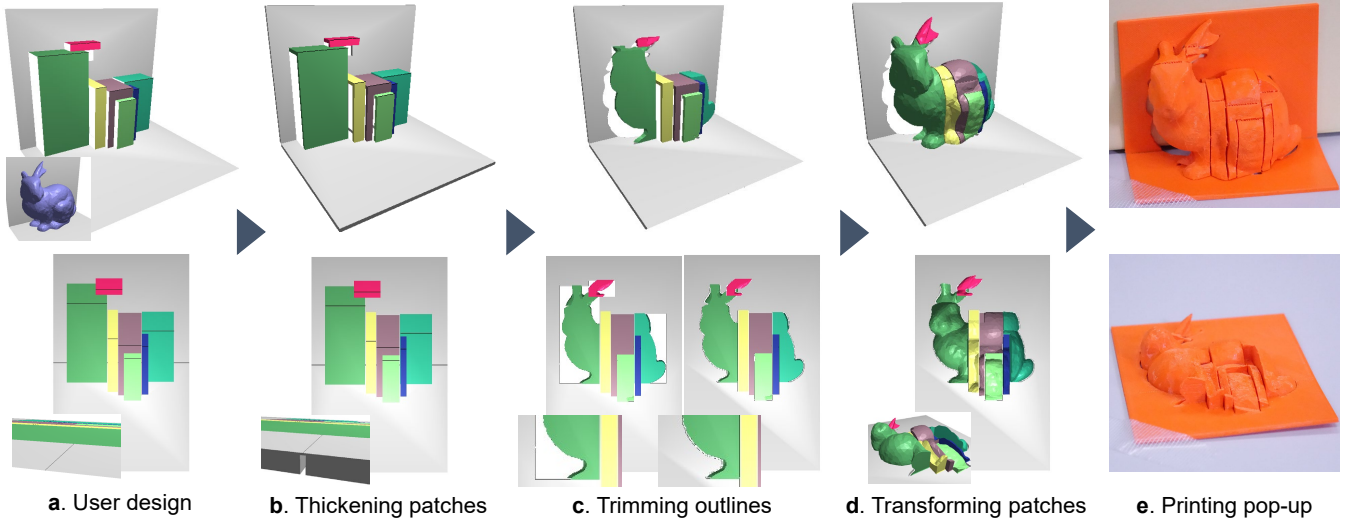
Figure 9: *Transformation procedure. Given the user-designed planner pop-up and the 3D model (a), we first add thickness to all patches (b). We next trim their outlines and fill the empty space (c). We then deform all patches using spatial segmentation (d), and finally fabricate the pop-up using a 3D printer (e).*

tained by sweeping the horizontal patch of C_i along the positive and negative y-axis, respectively. We denote the bounding box of S_i^{+x} and S_i^{+y} as B_i^+ and that of S_i^{-x} and S_i^{-y} as B_i^- (Figure 10e).

For a component C_i, we compute the region associated with its vertical patch $V(C_i)$ by subtracting, from its positive sweep region S_i^{+y}, all regions potentially interfering with it, as follows:

$$V(C_i) = S_i^{+y} \ominus \sum_{j \in N_i^y} (B_j^+ \cup B_j^-), \qquad (3)$$

where $\ominus$ is the CSG subtraction, $\cup$ the CSG union, Σ the CSG union of all elements, and N_i^y a set of components placed above C_i. Similarly, we compute the region associated with the horizontal patch $H(C_i)$ by subtracting, from its positive sweep region S_i^{+x}, all regions potentially interfering with it, as follows:

$$H(C_i) = S_i^{+x} \ominus \sum_{j \in N_i^x} (S_j^{+x} \cup B_j^-), \qquad (4)$$

where N_i^x is a set of components placed farther than C_i along the x-axis. Figure 10ab shows an example of spatial segmentation.

Notably, the two regions, $V(C_i)$ and $H(C_i)$, are defined with slightly different equations ((3) and (4)). If we define the two regions with symmetric equations,

$$V(C_i) = S_i^{+y} \ominus \sum_{j \in N_i^y} (S_j^{+y} \cup B_j^-), \ H(C_i) = S_i^{+x} \ominus \sum_{j \in N_i^x} (S_j^{+x} \cup B_j^-),$$

a region associated with different patches may be obtained (Figure 11c). Moreover, the following equations,

$$V(C_i) = S_i^{+y} \ominus \sum_{j \in N_i^y} (B_j^+ \cup B_j^-), \ H(C_i) = S_i^{+x} \ominus \sum_{j \in N_i^x} (B_j^+ \cup B_j^-),$$

generate regions associated with no components, even though they are available without interference (Figure 11d).

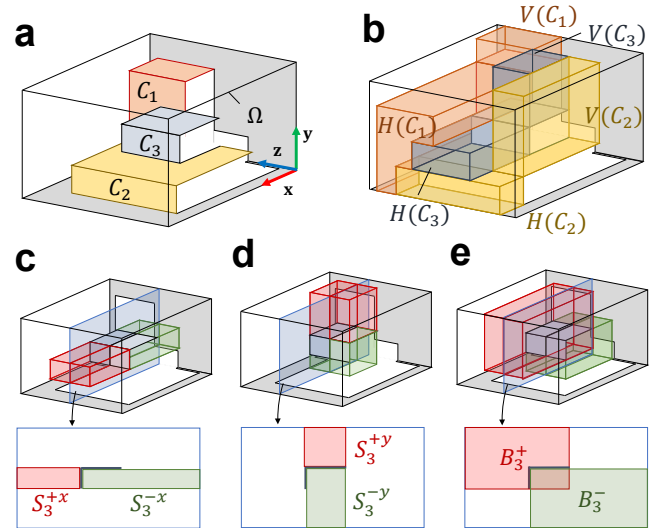After spatial segmentation, the component patches are deformed

Figure 10: *Example of spatial segmentation. Given a pop-up consisting of three components, C_1, C_2, and C_3 (a), we segment the entire space Ω so that two regions $V(C_i)$, and $H(C_i)$ are associated to two patches of each component C_i. For this spatial segmentation, we consider regions defined by sweeping a vertical and horizontal patch along the x- and y-axes (c, d) and their bounding boxes (e).*

using the intersection between the patch regions and the input model $\mathcal{M}$. The shapes of the vertical and horizontal patches are defined as $V(C_i) \cap \mathcal{M}$ and $H(C_i) \cap \mathcal{M}$, respectively. If the intersection contains isolated shapes, we remove them. Specifically, we perform connected component labeling to the intersection and remove disconnected shapes not in contact with C_i. Finally, we add
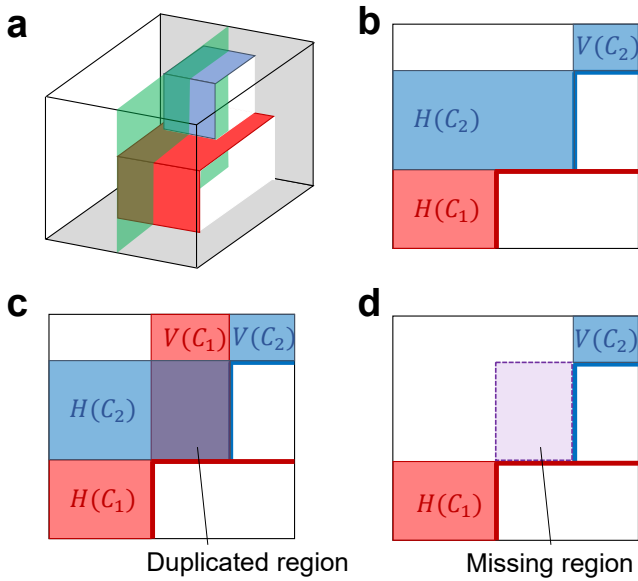
Figure 11: *Spatial segmentation with different formulations. From a component layout (a), two regions, $V(C_i)$ and $H(C_i)$, associated with vertical and horizontal patches of a component C_i are computed with our formulation (b), and with symmetric formulations (c, d).*

45° bevels at the convex fold and make gaps of width g at the concave and convex folds (Figure 4). We want to emphasize again that this patch deformation still ensures that the pop-up mechanism can be folded and unfolded without self-interference.

6. Results and Discussion

6.1. Modeling Fabricatable 90° Pop-ups

Figures 1 and 16 show fabricatable 90° pop-up models designed by authors. Each model was designed within 5–15 min of an interactive process. The models were fabricated in their unfolded states ($\theta = 180°$) using a 3D printer and could be converted to their folded states ($\theta = 90°$) without requiring additional assembly processes. The fabricatable pop-up models shown in Figure 16ceg were generated by placing multiple components parallelly onto the base patch. As the proposed method transforms component patches according to the input models, it can represent various models with such simple pop-up mechanisms. The models shown in Figure 16dfh were generated with hierarchical components. By arranging components hierarchically, it is possible to make pop-up structures approximating the input models more closely, resulting in a smaller height for each component.

With our method, it is possible to create pop-ups representing 3D objects with holes, such as torus, similar to the previous method [LLLN*14]. There are two approaches to place a hole in a pop-up model. The first is to place a pop-up component such that its patch covers the hole of the target 3D model (see Figure 12a). The second is to place a component C such that its associated region (i.e., $V(C)$ or $H(C)$) covers the hole of the target model (see Figure 12b).

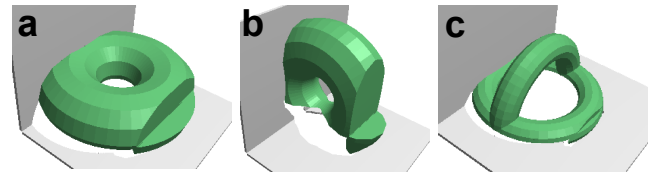By combining these approaches, we can model more complicated objects.

Figure 12: *Fabricatable 90° pop-ups created using target 3D objects with holes, such as torus (a, b) and a genus-2 object (c).*
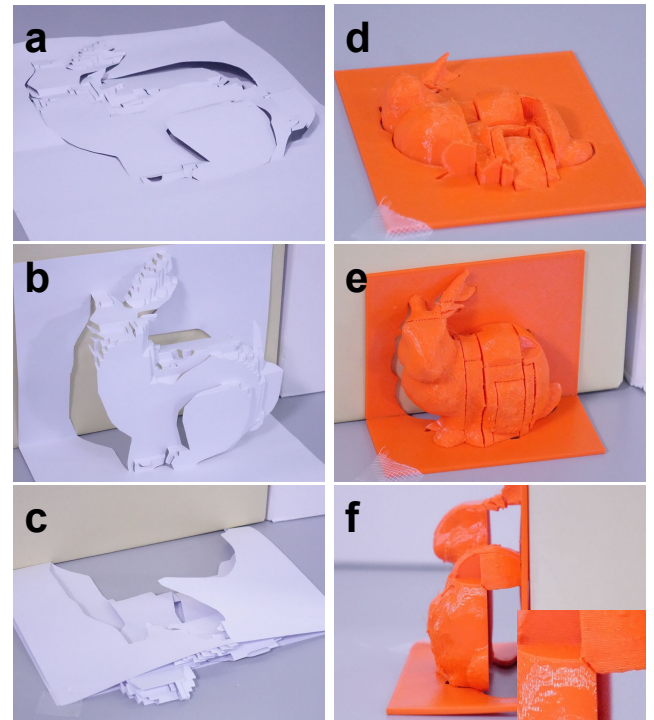

Figure 13: *Comparison between bunny models created with a paper-based technique [LSH*10] (left) and our method (right). While a paper pop-up consists of planner components, our fabricatable pop-up consists of non-planer components.*

Figure 13 compares bunny models generated with our method and a previous paper-based technique [LSH*10]. We created the paper pop-up bunny using a 2D pop-up template downloaded from the website of [LSH*10]. An advantage of our method over the previous paper-based techniques [LSH*10,LLLN*14] is the ability to generate pop-ups with non-planer patches. In this example, the shape of the bunny is better reconstructed by patches with curved shapes. Another advantage is the assembly-free creation process. In contrast, a disadvantage of our method is its size limitation; the maximum size of fabricatable 90° pop-ups is limited by the spec of a 3D printer. Creating large pop-ups with our method is more complicated than paper-based techniques. Another disadvantage is the restricted folding angle; while paper pop-ups can be completely

folded ($\theta = 0°$) due to the negligible thickness of the paper, the minimum folding angle of our pop-up models is 90°. This limitation arises because each patch has the inherent thickness and each convex fold has a 45° bevel (Figure 13f).

The proposed method has two main advantages in fabricating the models using an FDM 3D printer. First, the proposed method reduces the height of the models during printing by unfolding them. Generally, 3D printing processes are more stable when printing parts closer to the printing bed and less stable when printing parts farther away (higher) from the bed. Table 1 summarizes the heights of the fabricatable pop-ups in the unfolded state and the original models. We could reduce the height of all the models, making their fabrication stable. Second, the proposed method can reduce the support material. When a 3D model has overhang shapes, there is a need to place supports under them. The proposed method converts a model into a pop-up and unfolds it for printing, substantially reducing overhang shapes. Notably, each patch of the resulting pop-ups may have overhanging shapes, and the supports cannot be reduced to zero.

Table 1: *Height values (mm) of the original 3D models and fabricatable 90° pop-ups in the unfolded state, and their ratios.*

Model	Original (mm)	Ours (mm)	Ratio
Fig.16 a: Bunny	75.154	21.240	0.283
Fig.16 b: Dragon	66.748	32.890	0.493
Fig.16 c: Capitol	74.113	23.425	0.316
Fig.16 d: Tower	175.014	22.031	0.126
Fig.16 e: PG23	56.970	5.000	0.088
Fig.16 f: Building	163.968	17.846	0.109
Fig.16 g: Statue 1	61.255	23.193	0.379
Fig.16 h: Statue 2	154.142	27.437	0.178

6.2. User Study

To evaluate the usability of our method, we conducted a user study involving three university students who have no experience in 3D modeling. Initially, each participant went through a 20-minute tutorial; they first learned our prototype software and created a tower model similar to the one shown in Figure 16d using reference images. Subsequently, each participant was given a 3D model of a building as the target. They then freely modeled a fabricatable 90° pop-up until they were satisfied with a result.

Pop-up models created by three participants are shows in Figure 14. They placed 5, 6, and 12 components and took 17 min, 38 min, and 20 min to complete their models, respectively. Since our system supports the creation of pop-ups that meet the necessary constraints, all participants successfully created models with functional pop-up mechanisms. While one participant took additional time to fine-tune the components' position and size, they all completed the task in a reasonable time, especially considering their lack of modeling experience.

7. Conclusions and Future work

This paper proposes a method for designing and fabricating 90° pop-ups. In the proposed method, the user can interactively design a
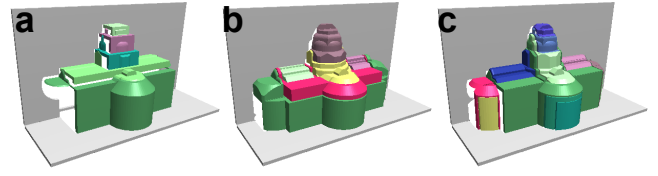

Figure 14: *Results of user study. Three participants created fabricatable 90° pop-ups with different component layouts (a-c).*

pop-up mechanism by placing the components such that the mechanism roughly approximates the target 3D model. Subsequently, all the components are deformed via CSG operations so that the entire mechanism better approximates the target model. The resulting pop-ups can be fabricated using a 3D printer, and their pop-up functions work without requiring additional assembly processes. We successfully fabricated various pop-ups using this method, indicating its feasibility.

One limitation of the proposed method is the existence of ignored regions. To prevent interference when folding and unfolding the pop-ups, we deform each patch in the region defined by sweeping it in the normal direction. Thus, parts of the input model outside the swept regions are ignored. In the example of Figure 15ab, we modeled a sphere with a single pop-up component and a quarter of it was ignored. In our future studies, we will develop a new component to reduce the ignored region. In traditional paper pop-ups, a concave component can be placed on the connection edge (Figure 15c). In our future studies, we would like to extend the proposed method to deal with such concave components. Another future direction is to automate the placement of the pop-up components.
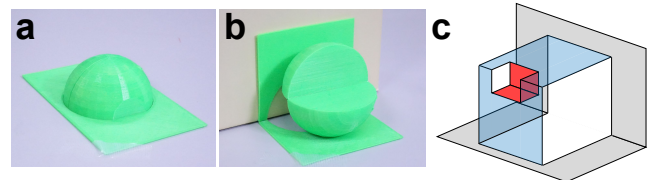

Figure 15: *Our method ignores object parts outside the patch-swept region (a, b). A concave component (red) can be placed in traditional paper pop-ups (c).*
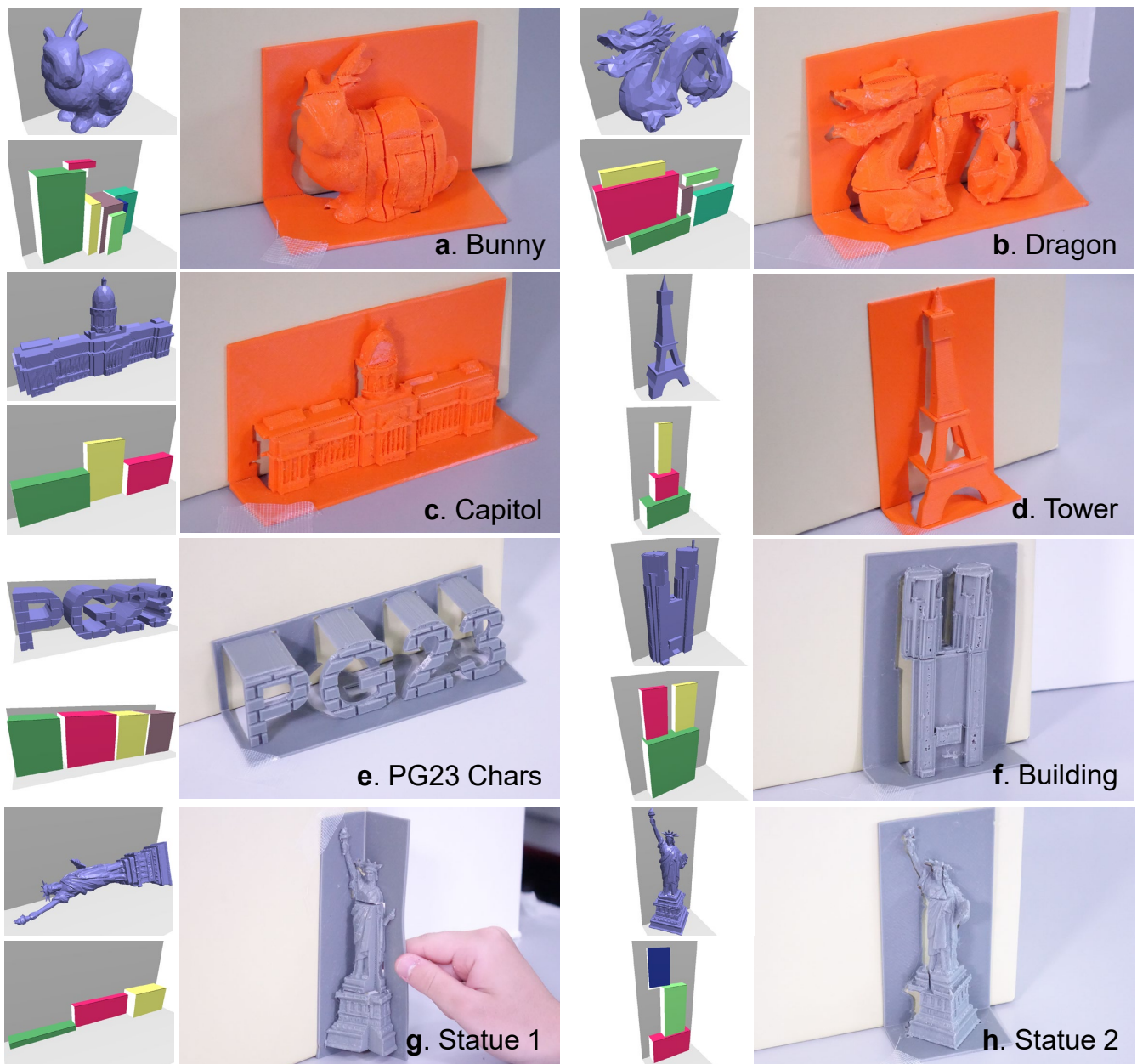
Figure 16: *Fabricatable 90° pop-ups created using the proposed method.*

References

[ATG*18] AN B., TAO Y., GU J., CHENG T., CHEN X. A., ZHANG X., ZHAO W., DO Y., TAKAHASHI S., WU H.-Y., ZHANG T., YAO L.: Thermorph: Democratizing 4d printing of self-folding materials and interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Apr. 2018), p. 260:1–260:12. doi:10.1145/3173574.3173834. 2

[BBO*10] BICKEL B., BÄCHER M., OTADUY M. A., LEE H. R., PFISTER H., GROSS M., MATUSIK W.: Design and fabrication of materials with desired deformation behavior. *ACM TOG 29*, 4 (July 2010), 63:1–63:10. doi:10.1145/1778765.1778800. 2

[Cha85] CHATANI M.: *Origamic Architecture Toranomaki*. Shokokusya, 1985. 1

[CTN*13] COROS S., THOMASZEWSKI B., NORIS G., SUEDA S., FORBERG M., SUMNER R. W., MATUSIK W., BICKEL B.: Computational design of mechanical characters. *ACM TOG 32*, 4 (July 2013), 83:1–83:12. doi:10.1145/2461912.2461953. 2

[DC15] DENG D., CHEN Y.: Origami-Based Self-Folding Structure Design and Fabrication Using Projection Based Stereolithography. *Journal of Mechanical Design 137*, 2 (02 2015). doi:10.1115/1.4029066. 2

[Gla02a] GLASSNER A.: Interactive pop-up card design. 1. *IEEE Com-*

puter Graphics and Applications 22, 1 (Jan.-Feb. 2002), 79–86. `doi:10.1109/38.974521`. 2

[Gla02b] GLASSNER A.: Interactive pop-up card design. 2. *IEEE Computer Graphics and Applications 22*, 2 (Jan.-Feb. 2002), 74–85. `doi:10.1109/38.988749`. 2

[HHR00] HUI E. E., HOWE R. T., RODGERS M. S.: Single-step assembly of complex 3-d microstructures. In *Proceedings of IEEE Thirteenth Annual International Conference on Micro Electro Mechanical Systems* (Jan. 2000), pp. 602–607. `doi:10.1109/MEMSYS.2000.838586`. 1

[Hul94] HULL T.: On the mathematics of flat origamis. *Congressus Numerantium 100* (Nov. 1994), 215–224. 2

[IEM*11] IIZUKA S., ENDO Y., MITANI J., KANAMORI Y., FUKUI Y.: An interactive design system for pop-up cards with a physical simulation. *Vis. Comput. 27*, 6-8 (June 2011), 605–612. `doi:10.1007/s00371-011-0564-0`. 2

[JDZ*22] JIAN B., DEMOLY F., ZHANG Y., QI H. J., ANDRÉ J.-C., GOMES S.: Origami-based design for 4d printing of 3d support-free hollow structures. *Engineering 12* (2022), 70–82. `doi:10.1016/j.eng.2021.06.028`. 2

[JLYL14] JR. C. R. R., LE S. N., YU J., LOW K.-L.: Multi-style paper pop-up designs from 3d models. *Computer Graphics Forum 33*, 2 (2014), 487–496. `doi:10.1111/cgf.12320`. 2

[KUS*21] KUSHNER S., ULINSKI R., SINGH K., LEVIN D. I., JACOBSON A.: Levitating rigid objects with hidden rods and wires. *Computer Graphics Forum 40*, 2 (May 2021), 221–230. `doi:10.1111/cgf.142627`. 2

[KWD*15] KWOK T.-H., WANG C. C. L., DENG D., ZHANG Y., CHEN Y.: Four-dimensional printing for freeform surfaces: Design optimization of origami and kirigami structures. *Journal of Mechanical Design 137*, 11 (Oct. 2015). `doi:10.1115/1.4031023`. 2

[LJGH11] LI X.-Y., JU T., GU Y., HU S.-M.: A geometric study of v-style pop-ups: Theories and algorithms. *ACM TOG 30*, 4 (July 2011), 98:1–98:10. `doi:10.1145/2010324.1964993`. 2

[LLLN*14] LE S. N., LEOW S.-J., LE-NGUYEN T.-V., RUIZ C., LOW K.-L.: Surface and contour-preserving origamic architecture paper pop-ups. *IEEE Transactions on Visualization and Computer Graphics 20*, 2 (Feb. 2014), 276–288. `doi:10.1109/TVCG.2013.108`. 1, 2, 3, 7

[LSH*10] LI X.-Y., SHEN C.-H., HUANG S.-S., JU T., HU S.-M.: Popup: Automatic paper architectures from 3d models. *ACM TOG 29*, 4 (July 2010), 111:1–111:9. `doi:10.1145/1778765.1778848`. 1, 2, 3, 7

[Mit09] MITANI J.: A design method for 3d origami based on rotational sweep. *Computer Aided Design and Applications 6*, 1 (2009), 69–79. `doi:10.3722/cadaps.2009.69-79`. 2

[MS04a] MITANI J., SUZUKI H.: Computer aided design for origamic architecture models with polygonal representation. In *Proceedings of Computer Graphics International, 2004.* (2004), pp. 93–99. `doi:10.1109/CGI.2004.1309197`. 2, 4, 5

[MS04b] MITANI J., SUZUKI H.: Making papercraft toys from meshes using strip-based approximate unfolding. In *Proceedings of ACM SIGGRAPH 2004 Papers* (Aug. 2004), p. 259–263. `doi:10.1145/1186562.1015711`. 2

[NKS*23] NARUMI K., KOYAMA K., SUTO K., NOMA Y., SATO H., TACHI T., SUGIMOTO M., IGARASHI T., KAWAHARA Y.: Inkjet 4d print: Self-folding tessellated origami objects by inkjet uv printing. *ACM TOG 42*, 4 (July 2023), 117:1–117:13. `doi:10.1145/3592409`. 2

[OI09] OKAMURA S., IGARASHI T.: An interface for assisting the design and production of pop-up card. In *Proceedings of Smart Graphics* (2009), pp. 68–78. `doi:10.1007/978-3-642-02115-2_6`. 2, 4

[PWLSH13] PRÉVOST R., WHITING E., LEFEBVRE S., SORKINE-HORNUNG O.: Make it stand: Balancing shapes for 3d fabrication. *ACM TOG 32*, 4 (July 2013). `doi:10.1145/2461912.2461957`. 2

[SA01] SONG G., AMATO N. M.: A motion planning approach to folding: from paper craft to protein folding. In *Proceedings of 2001 ICRA. IEEE International Conference on Robotics and Automation* (May 2001), vol. 1, pp. 948–953. `doi:10.1109/ROBOT.2001.932672`. 1

[STL06] SHATZ I., TAL A., LEIFMAN G.: Paper craft models from meshes. *Vis. Comput. 22*, 9 (Sept. 2006), 825–834. `doi:10.1007/s00371-006-0067-6`. 2

[Tac10] TACHI T.: Origamizing polyhedral surfaces. *IEEE Transactions on Visualization and Computer Graphics 16*, 2 (June 2010), 298–311. `doi:10.1109/TVCG.2009.67`. 2

[UKSI14] UMETANI N., KOYAMA Y., SCHMIDT R., IGARASHI T.: Pteromys: Interactive design and optimization of free-formed free-flight model airplanes. *ACM TOG 33*, 4 (July 2014), 65:1–65:10. `doi:10.1145/2601097.2601129`. 2

[XKM07] XU J., KAPLAN C. S., MI X.: Computer-generated papercutting. In *Proceedings of 15th Pacific Conference on Computer Graphics and Applications (PG'07)* (2007), pp. 343–350. `doi:10.1109/PG.2007.10`. 2

[YP21] YAN Z., PENG H.: Fabhydro: Printing interactive hydraulic devices with an affordable sla 3d printer. In *Proceedings of The 34th Annual ACM Symposium on User Interface Software and Technology* (Oct. 2021), p. 298–311. `doi:10.1145/3472749.3474751`. 2

[ZAC*17] ZHANG R., AUZINGER T., CEYLAN D., LI W., BICKEL B.: Functionality-aware retargeting of mechanisms to 3d shapes. *ACM TOG 36*, 4 (July 2017), 81:1–81:13. `doi:10.1145/3072959.3073710`. 2

[ZFO*22] ZHAO Z.-Y., FANG Q., OUYANG W., ZHANG Z., LIU L., FU X.-M.: Developability-driven piecewise approximations for triangular meshes. *ACM TOG 41*, 4 (July 2022), 43:1–43:13. `doi:10.1145/3528223.3530117`. 2