Folding Cartons: Interactive Manipulation of Cartons from 2D Layouts

Shuang Shan [1], Yiting Ma [1], Chengcheng Tang [2], Xuejin Chen [1]

[1] University of Science and Technology of China
[2] Stanford University

Abstract

Cartons are broadly used in the packaging industry and in our daily life. While most traditional packages are axis-aligned cuboids, there is a recent trend of creating paper packages and cartons with more complicated geometries. In this work, we present an interactive system to manipulate 3D cartons from 2D layouts. Two key components, *automatic folding* and *suggestive editing*, make our system remarkably easy to use. The automatic folding step computes the 3D coordinates of vertexes in a 3D carton model based on a simple observation. Then, the suggestive editing step determines several geometric constraints, such as vertex merging, merging propagation, and panel pasting, so that users can flexibly manipulate 3D shapes. Moreover, our system supports automatic updating of the corresponding 2D layouts when the user constructs and edits carton shapes in 3D to correct and explore potential layouts.

1 Introduction

Cartons have been widely used in the packaging industry to organize, ship, and deliver various commodities, which include food, daily necessities, and electronic components. Beyond the very basic packaging shapes like cuboids, extremely elaborated packaging exists for certain specialty items, such as wedding candies or wine bottles. The popularity of such intricate designs has increased, not to mention the fact that they are environmental friendly due to their recyclability and degradability [19].

Cartons are usually designed based on experience and trial-and-error. Designers typically start a package design by generating 2D vector artwork; then, a 3D mockup is essential for designers and clients to see the real appearance [8]. There are two ways to create a 3D mockup. A digital mockup is a great, inexpensive way of showing how the design would appear in real life, plus it allows designers to repeatedly modify the layout. A physical mockup is also helpful when making sure the size is correct in order to make a final decision. Due to the low cost and efficiency of digital mockups, there have been many software packages developed to help designers improve their design efficiency and productivity. For example, the commercial software named STUDIO [7] generates 3D models by manually assigning angles to folding edges. By using models, users can turn their ideas into beautiful 3D images. Moreover, it is non-trivial for non-expert users to figure out how to fold a complex carton using only a 2D crease pattern. A digital simulation system would provide valuable references to inspire creativity [30, 13, 20]. However, the existing software still requires extensive manual skills to design cartons. In order to create a virtual 3D model and explore diverse layouts, our idea is to fold the existing layout directly into a 3D model and achieve a variety of layouts by manipulating the corresponding model.

It is nontrivial to mathematically formulate the problem of model construction using only a planar layout, because the information extracted from the 2D pattern, such as edge length and vertex degree, may cause ambiguities on account of repeated edges and vertexes. Moreover, Biedl et al. [1] proved that given a polygon and a set of creases, it is NP-hard to know whether a polyhedron can be obtained by folding along the creases, which can also illustrate the difficulty of our problem.

In order to solve the folding cartons using 2D layouts as our only input, we present an optimization method when given a set of shape constraints. Moreover, an interactive design and exploration framework is proposed to allow users to visualize planar layouts and corresponding models in a 3D view and freely edit the models. The original layout can be corrected and explored by optimization. Figure 1 shows our folding results based on a series of 2D layouts designed by artists.

The contributions of this paper includes the following:

(1) An interactive and explorative system to manipulate the shapes of 3D cartons. Our system creates corresponding 3D models using rough initializations and user interactions. Imprecise layouts can be modified with a geometric optimization procedure. Users may easily edit these 3D models in order to explore alternative deformed layouts.

(2) A collection of shape constraints, which are represented through a set of points, are used to implement shape optimization. Based on observations of existing cartons, constraints are summarized by including panel rigidity and coplanarity to maintain the integrity of the cartons' shapes. In addition, computer-aided detection, like vertex merging and panel pasting, is introduced to offer suggestions to users.

Figure 1: Two cartons and artist-designed layouts. The corresponding 3D models are folded according to our system.

2 Related Work

2.1 Folding problem

Origami folding, which produces a 3D shape from a single planar sheet, has been widely studied for decades. Earlier techniques focus on simulating this folding process with a 2D crease pattern and a corresponding configuration of fold angles [30, 13, 20]. Typically, these simulation systems require the folding order and the angles of the creases to exist within a 2D pattern. If a 2D crease pattern is provided for a mountain or valley design, a continuous process can be simulated to design transformable and deployable structures [25, 26]. The above simulation systems can be applied primarily for paper origami with zero-thickness. Recently, many mechanical engineering techniques [27, 5, 14] have been proposed to develop the kinetic synthesis of thick panels that can be folded identically like zero-thickness origami. These simulation systems are limited to flat-foldable rigid origami with known mountain or valley crease designs. In comparison, our system only takes the 2D crease pattern as input without knowing the folding angles of the creases.

There are also some methods for solving related problems in carton folding. Song et al. [24] modeled foldable objects as tree-like multilink objects and used probabilistic roadmap methods [10] to find a sequence of motions to transform one configuration of a foldable object into another configuration. Mullineux et al. [19] provided a simulation framework for carton construction by considering geometric constraints for assembly. Both of these two methods require the target 3D state to be given as a premise, while our work aims to recover the 3D configuration of the folded carton from a 2D layout.

Compared with the straight creases in the above approaches, curved folding, which involves the use of curved folds in the crease pattern, has also drawn a lot of attention in computer graphics. Tang et al. [28] introduced a method for interactive design of developable surfaces that can handle side constraints including curved folding boundary conditions and isometry during interactive manipulation. Kilian et al. [11] presented an optimization-based framework to reconstruct a 2D development from a reference 3D surface. Solomon et al. [23] introduced a discrete paradigm to model developable surfaces based on 2D configurations, where the crease pattern and the corresponding crease angles are pre-defined. Recently, Kilian et al. [12] designed practical mechanisms to fabricate a curved folded surfaces simply by pulling a network of strings. By contrast, we focus on an inverse problem of recovering the 3D shape from a 2D planar pattern without knowing any 3D information.

2.2 3D reconstruction of line drawings

Although the aforementioned methods have the same goal of constructing 3D models from 2D layouts, including vertexes and edges, there is a one-to-one correspondence between the vertexes in the 2D line drawing and 3D model; however, our problem recovers many-to-one correspondences and the 3D construction process is different.

A line drawing is defined as a 2D projection of an object that contains its vertexes and edges. 3D interpretations of line drawings have been studied for a long time. Some researchers treat this task as an optimization problem. Marill [17] first proposed the principle of minimizing the standard deviation of angles (MSDA) to emulate the interpretation of line drawings as 3D objects. This new criterion was used by many other researchers soon after. Leclerc et al. [15] combined MSDA with the deviation from planarity as objective terms. Cao et al. [4] added a symmetry metric of 3D objects to obtain more complicated results. Other researchers tried to solve this problem from the point of view of information theory. Marill [18] minimized the description length of objects based on the idea that humans usually pick the simplest option from infinite possibilities when they see a line drawing. Shoji et al. [22] minimized the entropy of the angle distribution between line segments by utilizing a genetic algorithm. Later, the strategy of splitting and merging was used to reconstruct the line drawings of complex 3D objects [16, 31].

Different from line drawings, our input consists of the expanded structural layouts of 3D carton objects as displayed in 2D planes. The 3D carton model is constructed based on the folding procedure instead of the mathematical projection from 3D to 2D.

2.3 Geometrical shape optimization

In geometrical shape processing, many algorithms have been proposed to enforce shape constraints, which have been successfully used in interactive tools and physical simulation [2, 9]. Bouaziz et al. [3] unified a large variety of geometric constraints into one optimization framework, thus providing a simple and robust implementation. Poranne et al. [21] described an interactive method to manipulate and optimize polyhedral meshes under constraints with a linear-time algorithm. Tang et al. [29] solved constrained equations by employing a Newton-type method in a fast way, which yielded an interactive system to model meshes constrained by equalities and inequalities. Deng et al. [6] developed an interactive tool to explore architectural design with shape constraints. They also provided an optimization method to enforce hard constraints and soft constraints at the same time.

The studies mentioned above focus on optimizing existing 3D shapes with geometric constraints by proposing different optimization solutions. 2D layouts are not involved in their problems. In contrast, our goal is to recover the structure and topology from an expanded layout to form a 3D carton model. The geometric constraints are automatically detected and presented for users to explore during the folding process.

3 Overview

Figure 2 shows the overview of our algorithm. The input 2D layout of a carton consists of a set of cutting edges (solid lines) and folding edges (dashed lines). Given the 2D layout, an undirected graph is built, and a set of polygonal panels are extracted by finding the minimum cycles in the graph, as Figure 2(b) shows. Therefore, the 2D layout can be represented by a polymesh $\mathcal{L} = (V, E, P)$, where V is the set of vertexes, E is the set of all edges, and P is the set of panels. The edge set $E = E_c \cup E_f$, where E_c is the set of cutting edges, and E_f is the set of folding edges. To build a 3D carton model $\mathcal{M} = (V, E, P)$ from the 2D layout $\mathcal{L}$, which share the same topology, we compute the 3D coordinates of all of the vertexes in V.

However, it is not intuitive to analytically define the desired final 3D shape from a 2D layout. One possible alternative is to detect all the possible geometric constraints, such as vertex merging, panel parallelism, orthogonality between adjacent panels, and then integrate all these possible constraints together to form a large equation system. The challenge is that these local constraints can not adequately describe complicated and creative designs. Moreover, there are many ambiguities when detecting these constraints in a 2D layout, such as edges and panels that exist within the same shape. We propose a two-step algorithm based on the observation that humans usually fold edges at a right angle in order to obtain a rough shape, and then they merge the vertexes or edges to create a stable 3D carton. First, an initial 3D model (Figure 2(c)) is constructed based on a specific angle along each folding edge, as described in Sec. 3.1. The user can then manipulate and explore the various 3D shapes of the carton based on a series of suggested operations provided by our system. The final model is shown in Figure 2(e). The details of each step are introduced in the following sections.

Figure 2: Given a 2D layout (a), we first extract its 2D mesh (b) with different panels in different colors. By providing each folding edge with the same angle, which is $\pi/2$, an initial 3D model can be constructed (c). The final carton model (e) is built by a shape optimization technique based on a series of suggested geometrical constraints (d).

Figure 3: A rough 3D shape is obtained by folding the edges between adjacent panels by $\pi/2$ in the 2D layout (a). This is carried out in a breadth-first manner, starting from the panel with the maximal area. (b), (c), (d), and (e) are the different folding stages in the initialization step.

3.1 Shape Initialization

Initially, we assume that the 2D layout $\mathcal{L}$ lies on the XOY plane, so that each vertex $\mathbf{v}_i(x_i, y_i, z_i)$ in V has $z_i = 0$. Each panel in P has a normal $\mathbf{n}_i = (0, 0, 1)^T$. To construct the 3D model, we can compute the 3D coordinates of all the vertexes. When humans fold a carton, the general process begins by folding each edge at a rough angle and then connecting the closest vertexes to obtain a box-like shape. Instead of directly computing the vertex coordinates, it is natural to first fold the 2D layout by assigning a rotation angle to each pair of adjacent panels in order to obtain a rough shape. As seen in existing designs on the Internet, most of the traditional cartons are cuboids for the purpose of holding files or delivering daily supplies. Although there is a recent trend to design more complicated layouts to attract consumers, the shape of these unusual cartons is similar to traditional boxes, as their functions are still packaging commodities. Therefore, we choose $\pi/2$ as the initial value of the rotation angle at each folding edge. First, a panel graph of a layout is constructed, as shown in Figure 3 (a). Each panel is a node, and there is an edge between two panels if they are connected by a folding edge. All panels in the 2D layout are recursively folded in a breadth-first manner. Starting from the panel with the maximal area, our system folds all of its adjacent panels by rotating them around their connecting edge by $\pi/2$ and then continuing to the others. Figure 3 illustrates the folding process of a cuboid carton.

Traditional cartons in cuboid shapes can reach an ideal state. Figure 4 shows a group of results that are generated by simply folding adjacent panels by $\pi/2$. However, many complicated designs still need refinement, such as the four examples in Figure 5. If we take the hexagonal box as an example, the 3D model can be improved by simply snapping the small glue flaps to their nearby side panels to ensure that the carton is stable without any manual intervention. In most cases, it is common practice to attach a glued flap to a side panel in order to assemble a closed box. As a result, we develop a suggestive interface that automatically detects potential geometric modifications from the rough carton shape and provides these suggestions for users to explore.

3.2 Shape Refinement

Though not perfect, the initial 3D shape provides a good starting point for generating the final carton model. To further refine the 3D shape, the 3D coordinates of all the vertexes are then refined based on a set of shape constraints, such as vertex merging, panel

Figure 4: Four examples of carton models that can be fully automatically generated from 2D layouts by folding each edge at a fixed angle $\pi/2$.

Figure 5: Four examples of the carton models that need more shape improvement. While the shapes are not cuboid, using a fixed angle $\pi/2$ leads to non-closed shapes and loose structures.

(a) Vertex Merging (b) Panel Pasting

Figure 6: 3D shape refinement based on merging vertexes (a), or pasting panels (b). The locations where the 3D shape changes are circled in blue.

pasting, and so on. In this step, the coordinates of vertexes are chosen as our objective instead of angles on folding edges, mainly because that the geometric constraints in 3D shapes can be more simply and intuitively represented by 3D vertex coordinates.

A suggestive interface is provided to users to efficiently explore better carton shapes, as described later in Sec. 4. Once the user selects a suggested shape refinement operation, we optimize the 3D shape based on a series of geometric constraints. Given the current mesh with vertex positions that are defined as $\{\hat{\mathbf{v}}_i\}_{i=1}^N$, a new mesh with the same topology but new vertex positions $\{\mathbf{v}_i\}_{i=1}^N$ is computed according to the specified shape constraints. The geometric constraints can be classified into two groups, shape rigidity constraints and shape modification constraints. First, to keep the rigidity of each panel, the shape rigidity constraints, including panel rigidity and coplanarity, correspond to the similarity constraint and plane constraints described in [3]. Second, once the vertexes or panels are confirmed to be merged to modify the rough shape, more modification constraints are added. Each constraint is defined accordingly.

Panel rigidity. Each panel maintains its shape during the folding process. This constraint is defined by keeping the distance between any pair of points $\mathbf{v}_a, \mathbf{v}_b$ on the panel the same.

$$||\mathbf{v}_a - \mathbf{v}_b||^2 - ||\hat{\mathbf{v}}_a - \hat{\mathbf{v}}_b||^2 = 0. \tag{1}$$

Coplanarity. For each panel P_k, the coplanarity constraint specifies that all vertexes in this panel should always lie on a plane. We compute the sorted eigenvectors $\mathbf{U} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$ of the 3×3 covariance matrix $\mathbf{C}^T\mathbf{C}$ where $\mathbf{C} = \{\mathbf{v}_{kj}\}_{j=1}^{N_k}$, $\mathbf{v}_{kj}$ is the jth vertex among N_k vertexes in the panel. By removing the last column of $\mathbf{U}$, we implement the plane projection as [3] described.

If a set of shape modification operations, such as vertex merging and panel pasting, are selected, more constraints are added to optimize the vertex positions.

Merging vertexes. For any two vertexes, $\mathbf{v}_p$ and $\mathbf{v}_q$, that are selected to be merged as the same vertex, we have:

$$\mathbf{v}_p - \mathbf{v}_q = \mathbf{0}. \tag{2}$$

Panel pasting. If two panels, P_a and P_b, need to be snapped together, all the vertexes from these two panels should satisfy the same coplanarity constraint.

Typically, only a few shape modification operations are selected at each time; the above constraints form an under-constrained system to solve the new vertex coordinates. In order to find a feasible solution for the constrained problem, soft constraints are added to keep the original positions of the vertexes that are not relevant to the selected shape modification.

Irrelevant vertexes. Vertexes that are not in the same panel as any merged or pasted vertexes should remain in their original locations. We add these soft constraints by adding a small weight w, which is set to 0.001 in our experiments to the following equation:

$$w(\mathbf{v}_i - \hat{\mathbf{v}}_i) = \mathbf{0}. \tag{3}$$

Bouaziz et al. [3] proposed an efficient optimization method to combine all these shape constraints together. We directly use their C++ library, ShapeOp, to solve our shape optimization problem once a shape modification operation is performed. Figure 6 shows two examples of shape refinement by merging three vertexes and pasting three panels, respectively.

(a) Vertex Merging (b) Merging Propagation (c) Panel Pasting

Figure 7: Three different suggestions on shape modification. (a) A group of vertexes (marked in red) are detected as mergeable. $\mathbf{v}_a$ and $\mathbf{v}_b$ are close to each other, and one edge connected to $\mathbf{v}_a$ is the same length as one edge connected to $\mathbf{v}_b$ (green edges). $\mathbf{v}_c$ is also considered to be merged with $\mathbf{v}_a$ and $\mathbf{v}_b$ because $\mathbf{v}_c$ is close to $\mathbf{v}_b$ and one edge of $\mathbf{v}_c$ is the same length as $\mathbf{v}_b$ (blue edges). (b) Once the user confirms a suggested vertex merging operation, our system automatically proposes a symmetric group of vertexes that can also be merged (marked in blue). (c) If the panels marked in red are detected as being close to each other and having similar normals, our system suggests that these panels can be pasted together.

(a) (b)

Figure 8: Two examples of layout modification from 3D model editing. For each example, a box-like carton can be generated from the given 2D layout. Users can edit the shape of the 3D carton by dragging edges in our system. By enforcing the shape rigidity constraints, our system automatically updates the 2D layout.

4 Suggestive Interface for Shape Modification

For some creative designs, the rough 3D carton model generated based on our heuristic approach still needs some improvements or the user might edit the carton shape. To help common users refine the carton model, our system is equipped with a set of smart shape refinement operations, such as automatically detecting vertexes that can be merged, propagating modifications to similar groups, and providing suggestions for users to explore. Moreover, when the user edits a carton model in the 3D space, which is more intuitive, our system is capable of revising the 2D layout automatically.

4.1 Suggestive Interaction

Our system automatically detects a group of potential shape constraints to improve the 3D carton model. These guiding constraints are provided to the user for to quickly select and manipulate the carton shape. In our system, three types of shape refinement operations, vertex merging, merging propagation, and face pasting, are automatically detected.

Vertex merging. For irregular shapes, which consist of non-rectangular panels or non-perpendicular adjacent panels, some vertexes are close to each other after the initialization step. A practical carton model can be obtained by simply merging these nearby vertexes together if they have coherent edges. Any pair of vertexes, $\mathbf{v}_a$, $\mathbf{v}_b$, in different panels are detected as mergeable, if $\|\mathbf{v}_a - \mathbf{v}_b\| < \epsilon$, and there is an edge connecting to $\mathbf{v}_a$ that is the same length as one edge connecting to $\mathbf{v}_b$. ϵ is a distance threshold, and it is set as 50mm in our experiments. More vertexes can be merged together, as shown in Figure 7(a), by gradually adding one vertex $\mathbf{v}$ each time if $\mathbf{v}$ is mergeable with at least one vertex in the current list of vertexes to merge.

Merging propagation. Once the user makes a decision to merge a subset of vertexes, $\mathbb{V} = \{\mathbf{v}_i\}_{i=1,\ldots,K}$ based on our automatic suggestions or manual selections, our system will detect if another subset of vertexes $\mathbb{V}_b$ exists that is symmetric to $\mathbb{V}$ and propagate the merging operation to $\mathbb{V}_b$, as shown in Figure 7(b). A subset $\mathbb{V}_b$ is symmetric to $\mathbb{V}_a$ if $|\mathbb{V}_a| = |\mathbb{V}_b|$, and there is one-to-one correspondence between the vertexes in $\mathbb{V}_a$ and $\mathbb{V}_b$. Here, $|\mathbb{V}|$ is the number of vertexes in a vertex subset $\mathbb{V}$. Two vertexes, $\mathbf{v}_a$ and $\mathbf{v}_b$, are defined as corresponding vertexes if they have the same number of edges and one-to-one correspondence between the edge lengths.

Panel pasting. Merging vertexes is sometimes not enough to generate a desired model because of glue flaps, as shown in Figure 7(c). Many small flaps should be pasted to their adjacent side panels when making a carton; therefore, our system automatically detects panels to paste together. A pair of panels P_a and P_b is considered mergeable if their normals, $\mathbf{n}_a$ and $\mathbf{n}_b$,

Figure 9: More carton examples that can be automatically generated. For each example, four subfigures are shown from left to right: the input 2D layout, an intermediate stage during folding, and the final 3D shape from two views.

Figure 10: The complete process of generating a carton model with user interactions. From left to right: the input structural layout used to generate a flat polymesh, the initial 3D model, two vertex merging confirmations, one panel pasting, and the final 3D model.

satisfy $\mathbf{n}_a \cdot \mathbf{n}_b > 0.5$, and for each vertex in $\{\mathbf{v}_i^a\}_{i=1}^{N_a}$ of the panel P_a, $|(\mathbf{v}_i^a - \mathbf{v}_c^b) \cdot \mathbf{n}_b| < \epsilon_d$, where $\mathbf{v}_c^b$ is the centroid of the panel P_b, and $\mathbf{n}_b$ is its normal. When involving multiple panels, the detection strategy is the same as the vertex merging step.

4.2 2D Layout Refinement from 3D Editing

Our system is highly efficient for generating a 3D carton from an existing 2D layout. It also allows users to creatively edit the current 3D carton model by dragging an edge, and it simultaneously refines the 2D layout automatically according to the 3D editing. When the 3D mesh changes, the new shape constraints are transferred to the 2D layouts by maintaining the panel rigidity (Eq. 1). Since the layout remains in a 2D plane, the coplanarity is enforced by set all the vertexes in the layout to lie on the XoY plane. The irrelevant vertexes that do not lie on the moving edge have to stay at their original locations, as described in Eq. 3. Figure 8 shows two examples of automatic refinement of 2D layouts based on 3D editing.

5 Results and Discussion

Based on the two-step approach for folding a carton model, as well as the suggestive user interface, our system is natural and easy to use. We show a series of results of cartons in various shapes. A user study was also conducted to examine the effectivenss of our suggestive system.

5.1 Carton Results

For most cuboid cartons, their 3D models can be generated by folding each edge by $\pi/2$, as shown in Figure 4 and Figure 9. Nowadays, in the common design process, designers typically spend a great deal of time on manually creating 3D carton models in 3D modeling software. Our suggestive interface saves a significant amount of effort, so designers can focus more on designing the appearance of cartons.

For more complicated cartons, shape refinement is required. The entire process of creating a carton model from a 2D layout is illustrated in Figure 10. The final model is generated after a total of three clicks: two clicks to confirm vertex merging and one click to confirm panel pasting. Note that our system automatically detects these possible geometric editing options for users to select; therefore, the user can eliminate the tedious process of selecting edges or vertexes and assigning precise angles or positions to them. A more challenging example is shown in Figure 11. By initially folding each edge as $\pi/2$, our system first generates a cube inside the carton for the six square faces. Occasionally, our system fails to detect the mergeable vertexes with a small threshold ϵ to merge nearby vertexes; however, the user could manually merge two vertexes. Then, our system automatically

Figure 11: When given a 2D layout (a), our system generates an initialized result (b). After nine user interactions, including selecting merging vertexes and panel pasting, the final model is obtained as (d).

Table 1: Statistics of the number of edges N_{edge}, number of panels N_{panel}, and the number of user interactions $N_{interaction}$ for the examples shown in this paper.

Examples	Fig.9(a)	Fig.9(b)	Fig.9(c)	Fig.9(d)	Fig.10	Fig.11	Fig.12(a)	Fig.12(b)	Fig.12(c)	Fig.12(d)	Fig.12(e)	Fig.12(f)
N_{edge}	49	62	46	45	54	67	40	43	42	38	48	30
N_{panel}	13	13	11	13	14	19	11	13	13	13	12	11
$N_{interaction}$	0	0	0	0	3	9	1	4	1	3	3	3

detects two symmetric vertexes that also can be merged. Finally, a hexagonal carton is generated after a limited number of user interactions.

More carton models, which were generated using our system with a few user interactions, are shown in Figure 12. The statistics of the panel number, edge number, and number of user interactions are listed in Table 1. We can see that only a few user interactions are needed for a variety of shapes in order to confirm the system's suggestions.

5.2 User Study

We conducted a user study to examine the productivity of our system for constructing digital 3D mockups from 2D layouts and to analyze the effectiveness of our system in guiding non-expert users to fabricate physical mockups from 2D layouts. There are two experiments in our user study. In the first experiment, participants were given a brief introduction, and then they were asked to construct a digital 3D model from the same 2D layout using our system and the commercial software, STUDIO, respectively. For each participant, 10 examples were randomly selected from the 34 2D design layouts and presented to the participant for folding using our system and STUDIO. A *two-alternative forced choices* design was used, so the participant was asked to choose which of the two systems they would prefer to use after considering operation simplicity and modeling efficiency. Furthermore, participants were asked to rate the necessity of layout optimization in our system by using a scale from one to five, with five indicating that it is very necessary.

In the second experiment, participants were separated equally into two groups. One group was asked to fold a sheet of paper into a physical carton. The 2D layout was printed onto this piece of paper, and the user was guided by a video showing the folding sequence according to our system. Another group was asked to make a carton without the guide video. Each participant was assigned with the same 2D layout (Figure 9(a)), because it is not easy to imagine the final mockup from the 2D layout at first sight. The fabrication times were recorded for the two groups.

Based on the two experiments, our goal was to test the following hypotheses:

- **Hypo1:** Our system takes less time and effort to construct a digital 3D mockup than the traditional software.
- **Hypo2:** Our system provides novel and practical functions for generating diverse layouts.
- **Hypo3:** Our system is effective for guiding non-expert users to fabricate complex cartons from 2D layouts.

We consider our three hypotheses in turn. With respect to **Hypo1**, we collect the answers to the former two questions from 30 participants, all over eighteen years of age. Both male and female participants were included, and none had any prior experience with designing or folding cartons in computer software. The results are shown in Figure 13(a). The chart shows that 93% of participants prefer our system with regard to the operation simplicity, and 83% of participants prefer our system with regard to the modeling time. We also performed a paired-samples t-test at the level $\alpha = 0.05$ to compare the preference significance. This test shows that our system is significantly preferred by participants, rather than STUDIO, which is a typical representative of the traditional software that needs to assign folding angles to construct 3D shapes. Most participants voted for our system because it takes much less time to make a 3D model using our system. Take Figure 12(e) for example, participants usually spent much time adjusting the folding angles of the creases, while only three clicks are required in our system. On the other hand, one comment from the participants who preferred STUDIO stated that the operation in STUDIO only required them to select and assign angles, while our system required them to learn more operations to construct the digital 3D model.

With respect to **Hypo2**, after editing carton shapes as Sec. 3.2 described, 24 participants gave the highest score to our layout optimization function. In addition to exploring the diversity of 2D layouts as Figure 8 shows, our system can also adjust the imprecise panels on the 2D layout to reach an ideal model. Figure 13(b) shows the final models constructed by our system and STUDIO, respectively. As we can see, Panel 1 is higher than Panel 2 in the digital model constructed by STUDIO, because the 2D layout is not as precise as the designer would like. In comparison, our system is able to correct these design errors by merging the 3D vertexes in different panels, which are circled in red. This step optimizes the corresponding 2D layout.

Considering **Hypo3**, 38 participants (19 in each group) were asked to transform the 2D layout shown in Figure 9(a) into a physical carton. The comparison of the fabrication times with and without our guide video is shown in Figure 13(c). As we can see, most participants who did not watch the guide video spent more time, averaging 155 seconds, to fold a carton than the other

Figure 12: More complicated results with shape refinement. In each example, the first two images represent the 2D layout and the initial model, and the last two show the final model from different views. The middle images illustrate the intermediate shapes with different shape constraints, which are marked in red circles.

group, averaging 99 seconds. We also performed an independent-samples t-test at the level $\alpha = 0.05$ to compare the fabrication times. The results show that the guide video is a highly effective tool for non-expert users in fabricating complex cartons.

Limitations. Our suggestive shape optimization is very useful in a wide range of cases. Nevertheless, there are still some failure cases that our system can not deal with. Figure 14 shows a case that is challenging for our current system. The panels circled in red should be pasted together to produce a strong enforced panel. However, our system only ensures the coplanarity of the selected three panels, and as a result, the yellow panel is expanded in the opposite direction. Moreover, the pink back panel is orthogonal to the bottom panel after initialization, and the two red edges are not aligned. This can be solved by adding more editing tools, such as aligning edges, or assigning a folding angle for a specific edge, such as STUDIO.

6 Conclusion and Future Work

In this paper, we present an interactive modeling system to construct a 3D carton model from a 2D expanded layout. Based on the automatically folded rough model, a series of shape refinement suggestions are provided to intelligently optimize and edit carton models. This assists users so that they can produce desired models efficiently and productively. Despite these achievements, our current system could be further improved by adding more intelligent editing tools. For example, when a user moves a vertex, the translation could be propagated to other symmetric vertexes. We would also like to improve our automatic folding stage by employing a more comprehensive optimization formulation that considers both shape closure and stability. Moreover, this feature would be a boon to designers as it would support appearance editing within our system.

Acknowledgments

We thank the reviewers for their generous reviews, and Siyu Hu for stimulating discussions. Chengcheng Tang would like to thank the support of NSF grant IIS-1528025, a Google Focused Research Award, a gift from the Adobe Corporation, and a hardware donation from NVIDIA. This work was supported by the National Natural Science Foundation of China under No. 61472377 and 61632006.

(a) System Preference (b) Layout Refinment (c) Fabrication Time

Figure 13: (a) Compared with STUDIO, more participants prefer our system with regard to both the operation simplicity and modeling time. (b) Our system is able to automatically refine the inaccurate 2D layout, while STUDIO only generates the folded shape based on user assigned angles. (c) Comparison of fabrication times with/without video. It took much longer for common users to fabricate a carton from a 2D layout without our guide.

Figure 14: Limitations in our current system. From left to right: a 2D layout, the initial model after folding each edge by $\pi/2$, the optimized model after pasting the three panels circled in red, and the desired carton model constructed from the 2D layout.

References

[1] T. Biedl, A. Lubiw, and J. Sun. When can a net fold to a polyhedron? *Comput. Geom. Theory Appl.*, 31(3):207–218, June 2005.

[2] M. Botsch, M. Pauly, M. Gross, and L. Kobbelt. Primo: Coupled prisms for intuitive surface modeling. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, pages 11–20, 2006.

[3] S. Bouaziz, M. Deuss, Y. Schwartzburg, T. Weise, and M. Pauly. Shape-up: Shaping discrete geometry with projections. *Comput. Graph. Forum*, 31(5):1657–1667, Aug. 2012.

[4] L. Cao, J. Liu, and X. Tang. 3D object reconstruction from a single 2D line drawing without hidden lines. *Proceedings of the Tenth IEEE International Conference on Computer Vision*, 01:272–277, 2005.

[5] Y. Chen, R. Peng, and Z. You. Origami of thick panels. *Science*, 349(6246):396–400, 2015.

[6] B. Deng, S. Bouaziz, M. Deuss, A. Kaspar, Y. Schwartzburg, and M. Pauly. Interactive design exploration for constrained meshes. *Computer-Aided Design*, 61:13 – 23, 2015.

[7] ESKO. Studio. https://www.esko.com/en/products/studio.

[8] G. Fussell. The beginner's guide to packaging design. https://design.tutsplus.com/articles/the-beginners-guide-to-packaging-design--cms-27380.

[9] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. In *ACM SIGGRAPH 2005 Papers*, pages 1134–1141. ACM, 2005.

[10] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, Aug. 1996.

[11] M. Kilian, S. Flöry, Z. Chen, N. J. Mitra, A. Sheffer, and H. Pottmann. Curved folding. *ACM Trans. Graph.*, 27(3):75:1–75:9, Aug. 2008.

[12] M. Kilian, A. Monszpart, and N. J. Mitra. String actuated curved folded surfaces. *ACM Trans. Graph.*, 36(3):25:1–25:13, May 2017.

[13] N. Kishi and Y. Fujii. Origami, folding paper over the web. In *Proceedings of the Third Asian Pacific Computer and Human Interaction*, pages 337–342. IEEE Computer Society, 1998.

[14] J. S. Ku and E. D. Demaine. Folding flat crease patterns with thick materials. *Journal of Mechanisms and Robotics*, 8(3), June 2016.

[15] Y. G. Leclerc and M. A. Fischler. An optimization-based approach to the interpretation of single line drawings as 3D wire frames. *International Journal of Computer Vision*, 9(2):113–136, 1992.

[16] J. Liu, Y. Chen, and X. Tang. Decomposition of complex line drawings with hidden lines for 3d planar-faced manifold object reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:3–15, 2010.

[17] T. Marill. Emulating the human interpretation of line-drawings as three-dimensional objects. *Int. J. Comput. Vision*, 6(2):147–161, June 1991.

[18] T. Marill. Why do we see three-dimensional objects? *Technical Report AI-Memo 1366*, 1992.

[19] G. Mullineux and J. Matthews. Constraint-based simulation of carton folding operations. *Comput. Aided Des.*, 42(3):257–265, Mar. 2010.

[20] R. Nimnual and S. Suksakulchai. Virtual reality for packaging folding practice. In *International Conference on Control, Automation and Systems*, pages 1011–1014, 2007.

[21] R. Poranne, E. Ovreiu, and C. Gotsman. Interactive planarization and optimization of 3D meshes. *Computer Graphics Forum*, 32(1):152–163, 2013.

[22] K. Shoji, K. Kato, and F. Toyama. 3-D interpretation of single line drawings based on entropy minimization principle. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, Apr. 2001.

[23] J. Solomon, E. Vouga, M. Wardetzky, and E. Grinspun. Flexible developable surfaces. *Comput. Graph. Forum*, 31(5):1567–1576, Aug. 2012.

[24] G. Song and N. M. Amato. A motion-planning approach to folding: from paper craft to protein folding. *IEEE Transactions on Robotics and Automation*, 20(1):60–71, Feb. 2004.

[25] T. Tachi. Simulation of rigid origami. *Proceedings of International Meeting of Origami Science Math and Education*, pages 175–187, 2009.

[26] T. Tachi. Geometric considerations for the design of rigid origami structures. In *Proceedings of the International Association for Shell and Spatial Structures Symposium*, pages 458–460, 2010.

[27] T. Tachi. Rigid-foldable thick origami. *Fifth International Meeting of Origami Science*, pages 253–264, 2011.

[28] C. Tang, P. Bo, J. Wallner, and H. Pottmann. Interactive design of developable surfaces. *ACM Trans. Graph.*, 35(2):12:1–12:12, Jan. 2016.

[29] C. Tang, X. Sun, A. Gomes, J. Wallner, and H. Pottmann. Form-finding with polyhedral meshes made simple. *ACM Trans. Graph.*, 33(4):70:1–70:9, July 2014.

[30] J. M. Thiel. Interactive manipulation of virtual folded paper. Master's thesis, UBC, 1998.

[31] C. Zou, H. Yang, and J. Liu. Separation of line drawings based on split faces for 3D object reconstruction. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 00:692–699, 2014.