Computer-Aided Design 114 (2019) 215–223

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Cartonist: Automatic Synthesis and Interactive Exploration of Nonstandard Carton Design☆,☆☆

Lifeng Zhu[a,b,*], Benyi Xie[a], Yongjie Jessica Zhang[b], Lap-Fai Yu[c]

[a] The State Key Laboratory of Bioelectronics, Jiangsu Key Lab of Remote Measurement and Control, School of Instrument Science and Engineering, Southeast University, China
[b] Department of Mechanical Engineering, Carnegie Mellon University, USA
[c] Department of Computer Science, George Mason University, USA

A R T I C L E I N F O

Article history:
Received 14 October 2018
Received in revised form 13 March 2019
Accepted 24 April 2019

Keywords:
Carton design
Shape space exploration
Customized fabrication

A B S T R A C T

Cartons are commonly used in the packaging industry, and standard carton patterns are broadly used in mass production. However, their designs may not fit different containees or specific fabrication requirements well. Taking the dimensions of the containees as input, we study the patterns of cartons and propose a system named "Cartonist" to efficiently explore the pattern variations. A synthesis algorithm is proposed to create a space of nonstandard carton patterns, based on which a design space exploration method is demonstrated. Cartonist enables a user to customize the design of carton patterns for different applications, considering design criteria such as material efficiency, folding ease and stability. We perform a user study to evaluate the effectiveness of our proposed system for carton design.

1. Introduction

Cartons are containers created by folding paperboards. They have been broadly used in the packaging industry for years. Products ranging from daily necessities to electronic commodities are commonly stored in cartons during their shipment, delivery and sale. Among various design of cartons, cubic cartons are mostly used in the packing industry. They are suitable for packing products, as well as stacking for reducing the cost and preventing failure in the transportation. Although cartons usually take the shape of a cuboid, multiple carton patterns can be folded into a carton with the same shape. The space of the carton pattern has a large variation in both topology and geometry. Moreover, a valid pattern may have several constraints and implicit design intentions, which also complicate the design problem. Fig. 1 depicts a carton example with three different patterns. It has not been well studied how to generate patterns and how to select them with specific design intentions.

In the traditional packaging industry, to save cost by mass production, only limited types of carton patterns with standard design parameters are fabricated. It is not flexible to use standard carton patterns when the producers are short of the cardboards or the packed products are changed. If non-standard carton patterns can be conveniently designed, it will be more flexible for the producers to adjust to the fluctuations in the supply chain. In recent years, thanks to the development of programming tools and customized fabrication devices such as 3D printers and laser cutters, fabricating non-standard products becomes easier and more affordable. This trend brings more opportunities to the packaging industry and calls for more powerful computational design tools for designing nonstandard carton patterns. Therefore, a system capable of generating a variety of carton patterns and allowing users to efficiently explore a target design is highly desired in the age of customized fabrication.

In this paper, we propose a system named "Cartonist" to support the design of nonstandard carton patterns. We analyze the typical carton patterns and propose a procedural method to randomly generate a variety of carton patterns. We then define several efficient metrics to evaluate the generated patterns, which encode typical design considerations in the packaging industry. Finally, we develop the Cartonist system which allows a user to efficiently explore the design space by specifying the design considerations.

In the following, we briefly review the related work in Section 2. Then we discuss the details of how we automatically create and explore the carton patterns in Section 3. We include

☆ This paper has been recommended for acceptance by H. Qin.

☆☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.cad.2019.04.007.

* Corresponding author at: The State Key Laboratory of Bioelectronics, Jiangsu Key Lab of Remote Measurement and Control, School of Instrument Science and Engineering, Southeast University, China.

E-mail addresses: lfzhulf@gmail.com, lifengzhu@seu.edu.cn (L. Zhu).

https://doi.org/10.1016/j.cad.2019.04.007

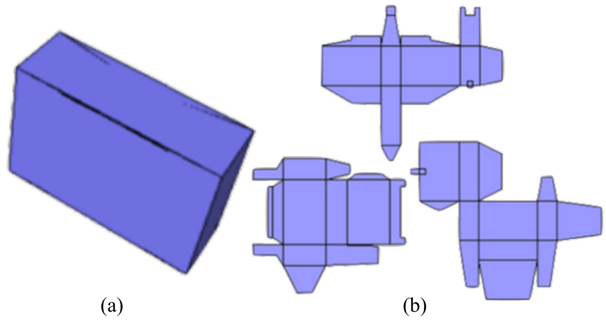


Fig. 1. A carton (a) and three different patterns (b) that can be folded into the same carton as shown in (a).

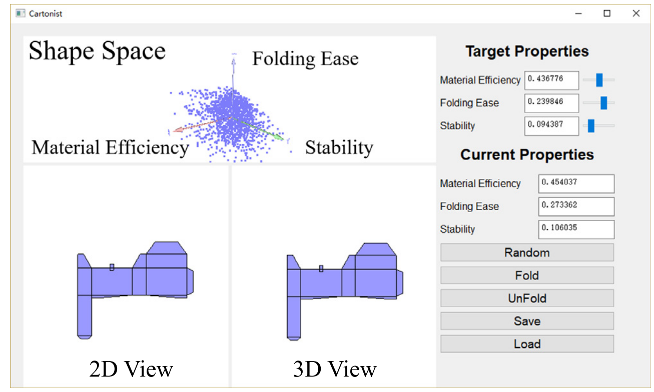


Fig. 2. The user interface of Cartonist. The bottom-left subwindow shows the 2D view of the unfolded carton. The bottom-right subwindow displays the 3D view of the folding animation. The top subwindow shows the shape space of synthesized samples of carton patterns in terms of three properties: the material efficiency, folding ease and stability, which are controlled by the slider bars on the top right. The text boxes show the properties of the carton design in the 2D view and 3D view. We also provide buttons that allow the user to randomly get a valid carton design, to fold or unfold the pattern step by step, as well as to save or load a design.

experiments for demonstrating our Cartonist system in Section 4 and conclude our work in Section 5.

2. Related work

Carton design is an application of general origami design. As the geometry of a two-dimensional (2D) folding pattern directly relates to its three-dimensional (3D) folded shape, there are lots of geometric constraints on origami design [1]. Many techniques have been developed to simulate the folding process of a designed origami [2,3]. Carton folding has also been studied in the areas of robotics and mechanics because it is related to the fabrication of packages. Constraints were modeled to simulate carton folding [4], and the assembly process was taken into consideration to model the folding sequence of cartons [5,6]. These studies assume that the carton patterns are given, while we focus on designing the carton patterns in this paper.

With the help of carton folding simulations, users can intuitively visualize the folded shape and use the feedback to interactively adjust their carton design [7,8]. Various pieces of computer-aided design (CAD) software, such as AristoCAD, support this kind of interactive design for cartons. To make the design process handier, Shan et al. [9] proposed operations such as vertex merging and panel pasting to help correct the inconsistency between the 2D pattern and the 3D folded carton. Vitalii et al. [10] presented a method to help users design the adjacency graph of the 2D pattern. Beyond the low-level geometry of carton designs, the affordance of packages was also considered in the carton design [11]. In this paper, we present a procedural model to create a substantial number of design samples and propose to explore the design space with regard to several high-level design properties.

In the shape modeling community, the idea of constructing a shape space and having users explore designs in the space has been proposed for years. To name a few, there are existing works allowing users to explore the shape spaces for designing furniture [12], garments [13], architecture surfaces [14] and layouts [15]. Adriana et al. [16] proposed to use hierarchical B-splines to construct the parametric space of CAD models with its simulated features. The problem of constructing shape spaces to support interpolation between two models with different topologies was also studied recently [17–19]. However, it is still challenging to represent the shape space for a large number of models with different topologies. In this paper, we synthesize a simple strategy to build the shape space of carton designs.

3. Method

We propose Cartonist, a system which aims to help the user quickly find feasible carton designs by specifying a few high-level requirements. In this paper, we define three high-level requirements to showcase Cartonist: the material efficiency in fabrication, the folding ease and the stability of the folded carton. We focus on cubic cartons in this paper because they are the most typical and common shapes in the packaging industry. Our system takes the dimensions of the cubic carton as input, i.e., the length, width and height of the folded carton. Its dimensions can be either manually specified or computed from the bounding box of the objects to be packed. As shown in Fig. 1, there are multiple valid designs of carton patterns that can be folded into the same carton.

The user interface of Cartonist is shown in Fig. 2. We quantify and normalize the high-level requirements, and include slider bars for the user to adjust his design preferences. As the user updates his design preferences, Cartonist automatically generates a carton design whose features are close to the user specifications. We show the 2D pattern in the bottom-left subwindow and its corresponding folding process in the bottom-right subwindow.

As shown in Fig. 4, our interactive design interface is driven by a system composed of an offline pattern synthesis stage and an online shape exploration stage. We first randomly generate samples of valid carton patterns and compute their corresponding indicators in the offline stage. Then we create a shape space of valid carton patterns and allow the user to explore a desired design in this space. We explain details of Cartonist in the following subsections.

3.1. Carton pattern synthesis

In our procedural technique for synthesizing carton patterns, the input includes the dimensions of the carton box and the output is a valid carton pattern which can be folded into the carton box with the target dimensions. By looking into the common features of carton patterns, we design a procedural modeling approach to synthesize the valid 2D patterns. Both the topological structure and geometric parameters are taken into consideration. We decompose our procedural modeling approach into five steps, and the intermediate result of each step is shown in Fig. 3.

Select Basic Trunk (Step 1). At the first step, we choose a basic trunk (Fig. 3(a)), which can be folded into a box without any sealing. In order to increase the variability of the 2D patterns, we include all possible unfolded patterns of a cube in Cartonist. As studied in discrete mathematics [20], there are only eleven

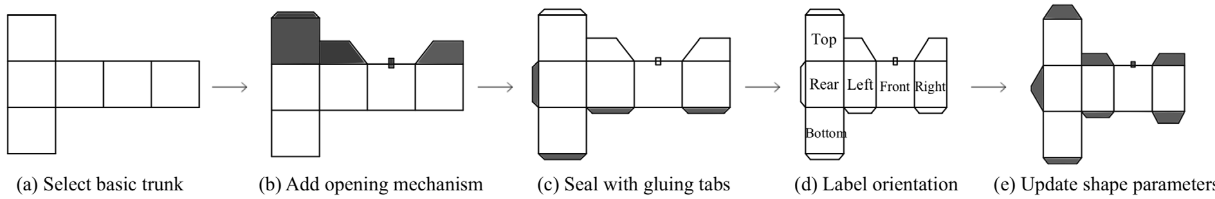


Fig. 3. A procedural model for synthesizing carton patterns, including five steps: (a) a basic carton trunk is selected; (b) opening mechanisms (highlighted in gray) are added to the trunk; (c) gluing tabs (highlighted in gray) are added to ensure all edges sealed; (d) the orientation of the carton (labeled in each face) and the dimension of the box are determined; and (e) the shape parameters of each carton mechanism (highlighted in gray) are updated . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

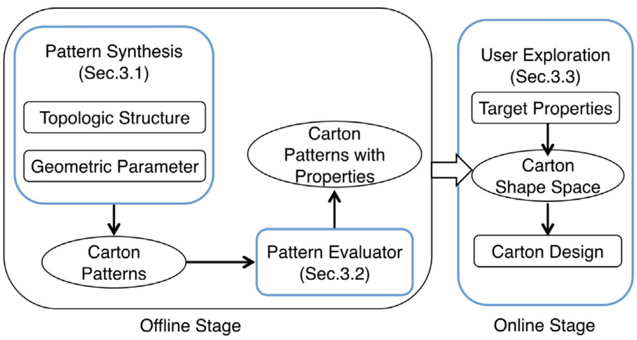


Fig. 4. A system overview of Cartonist.

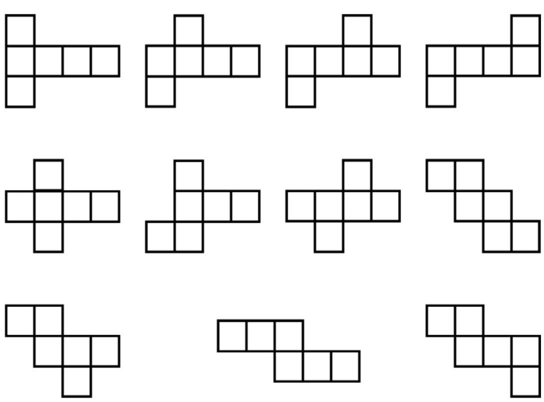


Fig. 5. Eleven basic trunks that can be folded into a cube.

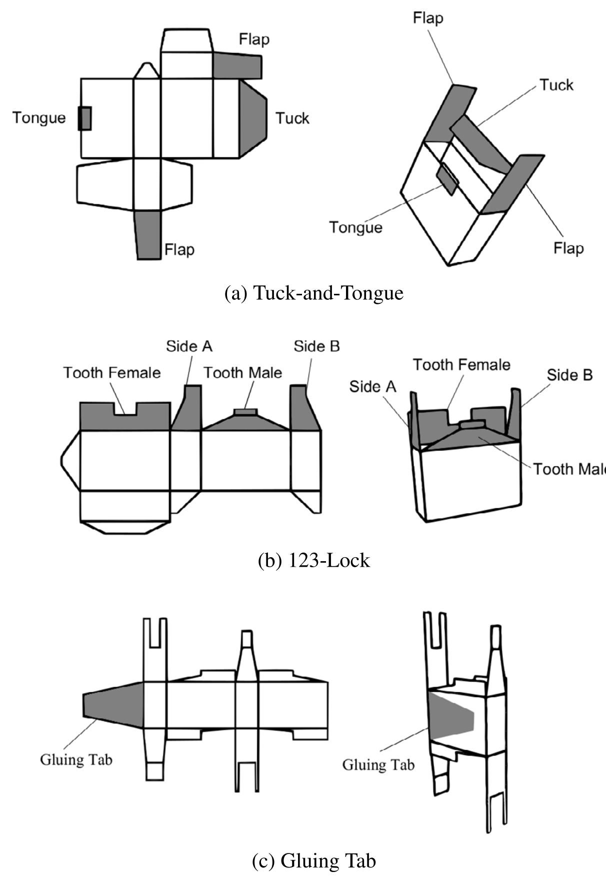


Fig. 6. Three typical opening mechanisms implemented in Cartonist.

unfolded patterns of a cube, which correspond to the eleven distinct pairings of the six-node trees as shown in Fig. 5. Therefore, we encode these eleven patterns as the candidates for the basic trunk. In our implementation, we store the lists of vertices, edges and faces for each pattern, as well as the indices of the edge pairs $P = \{(e_{i1}, e_{i2})\}$ that will be sealed to form a cube. We suppose e_{i1} is the sealing edge of e_{i2} and vice versa. Such connectivity information will be used in the following steps. In this step, we randomly choose one out of the eleven patterns to be the basic trunk.

Add Opening Mechanism (Step 2). After the basic trunk is selected, we then synthesize opening mechanisms onto the trunk. In Cartonist, we implement three typical mechanisms: tuck-and-tongue, 123-lock and gluing tabs [21], as shown in Fig. 6. The tuck-and-tongue mechanism includes a tuck, two flaps, and sometimes a tongue. After two flaps are folded to cover the containees, the tuck is inserted to close the carton by friction or glues, or locked by inserting a tongue. The 123-lock mechanism is also referred as Houghland Bottom, which is usually folded to lock the bottom by inter-overlapping four panels named tooth female, tooth male and two sides in the figure. A gluing tab is another common sealing mechanism. By attaching a panel on the boundary edge and gluing it onto another face, it seals a carton box.

We also parameterize the mechanisms and store the constraints on the parameters, which will be used to create the 2D embedding after the topology of the pattern is chosen. For a mechanism m, we define a set of parameters $\{x_{mi}\}$, the range of the parameters $\{l_{mi}, u_{mi} \mid l_{mi} \leq x_{mi} \leq u_{mi}\}$, as well as other constraints on the parameters $\{C_{mi}(x_{m1}, x_{m2}, \ldots) \geq 0\}$. Details of the parameterized model are listed in the Appendix. Note that we also support the traditional tuck mechanism as it is a special case when the size of the tongue vanishes. In this work, we assume a zero thickness of the planar material. In case the thickness cannot be ignored, we need to update the constraints by adding the thickness of the cardboard into the equations. Because the thickness of the planar material is much smaller compared with the dimensions of the carton box, we do not model it here.

In this step, we have to add a tuck-and-tongue mechanism or 123-lock mechanism to the trunk, as they are easy to open compared to gluing tabs. We randomly select a face f on the trunk with only one inner edge e_i, and the inner edge will act as a hinge when we open the carton. With the specified inner edge e_i, we then add the tuck to the edge e_j on the face f which is opposite to e_i, and the tongue will be added on the edge e_k, where the edge e_k stitches with the edge e_j after the pattern is folded. If we add a 123-lock mechanism to the face f, we delete f on the trunk and randomly put the four patches of the 123-lock on the inner edge e_i and the sealing edges of the other three edges on f. We then check whether the opposite face of f also has only one inner edge. If so, we can randomly add another tuck-and-tongue or 123-lock mechanism to the opposite face.

Seal with Gluing Tabs (Step 3). After the opening mechanism is added, we check whether all sealing edge pairs are associated with an opening mechanism. If there are any edge pairs with no mechanism, we randomly add a gluing tab to one edge in the pair. After this step, the carton will be sealed by one of the mechanisms in the end.

Label Orientation (Step 4). We set the orientation of the carton pattern by labeling all faces in the trunk with top, bottom, front, rear, left and right. By default, we set the face with a tuck-and-tongue mechanism as the top side and the face with the tongue as the front side, because it is common to open the carton with a tuck-and-tongue mechanism on the top from the front side. After the top and front sides are set, the labels of all other faces are uniquely defined by checking their adjacency relationship. If there is no tuck-and-tongue mechanism, we treat the 123-lock as the opening mechanism and set its face as the top side. Then we randomly assign one of its adjacent faces after folding as the front side and determine other orientation labels after the top and front sides are set. Till now, all the necessary topological information is generated. With the generated orientation labels, we resize the edge length on the trunk to fit with the input dimensions of the cube, i.e., its length, width and height.

Update Shape Parameters (Step 5). Finally, we generate the geometric information of the carton pattern by setting the parameters of each mechanism. For each opening mechanism, as the geometry of the trunk is defined, the range and constraints of the parameters are known in this step. We randomly generate parameters x_{mi} in its range $[l_{mi}, u_{mi}]$. If its parameters fall out of the constraint space $\{C_{mi}(x_{m1}, x_{m2}, \ldots) \geq 0\}$, we run the random number generator again until the parameters correspond to a valid mechanism. After the parameters are set, we get the length of each edge on the pattern and compute the position vector of the vertices by solving a Poisson equation with one arbitrarily fixed vertex [22]. Because we have known the direction of each edge vector after Step 3 and its corresponding edge length in this step, we obtain each edge vector $\mathbf{e}_{ij}$ which connects the ith and jth vertices. The edge direction is defined to align with the basic trunks shown in Fig. 5 and it cancels the rotational degree of freedom in the Poisson equation. Suppose the first vertex in the 2D pattern is fixed at the origin as $\mathbf{p}_0 = \{0, 0\}$, we solve

$$\begin{cases} \mathbf{p}_i - \mathbf{p}_j = \mathbf{e}_{ij} & \text{for all edges } ij \\ \mathbf{p}_0 = \{0, 0\} \end{cases}$$

to get the positions $\mathbf{p}_i$ of all other vertices on the 2D pattern.

We also run a self-intersection test after the 2D embedding of the pattern is obtained and discard those self-intersecting samples. Specifically, each face on the pattern is triangulated into a few triangles and we use the triangle-triangle intersection test to check whether there are overlapping faces on the 2D pattern. The carton pattern only has a few faces. Note that faces belonging to the same opening mechanism do not have intersection between them, so do faces on the basic trunk. Therefore, we further reduce the computation by ignoring the corresponding intersection tests. Because the constraints on the shape parameters guarantee that the opening mechanisms do not penetrate the basic trunk of the carton box and the opening mechanisms are designed to be intersection-free during folding, we do not run the self-intersection test during the folding simulation. If the constraints are not well modeled for irregular cartons, we may also run the self-intersection test during simulation.

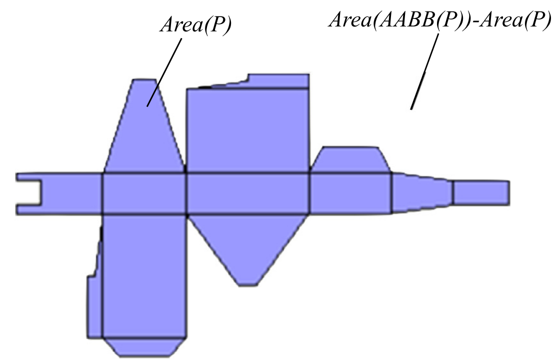


Fig. 7. Definitions of areas for computing the material efficiency and stability of a carton pattern P.

3.2. Evaluation of carton patterns

In order to incorporate high-level design considerations into our system, we evaluate each generated carton pattern. By introducing metrics to quantitatively evaluate each carton pattern, we are able to distinguish the patterns even if they all can be folded into the same cuboid. In Cartonist, we consider three properties of the carton: material efficiency, folding ease and stability. It is possible to design and incorporate other metrics. As long as we can map the carton pattern to a scalar measurement, Cartonist can be used to guide the design of carton patterns with respect to a particular design consideration.

Material Efficiency. Wastage is the first design consideration of our Cartonist system, as reflected by the material efficiency during the fabrication process. Cartons are typically formed by folding 2D patterns, which are usually obtained by using NC machining on plates of cardboard. It is preferable to have the 2D pattern well layout on the 2D material space. In this case, after cutting the pattern from the material, we expect to have less material wasted. We assume the planar material is a rectangular cardboard, and therefore the wastage is defined as the ratio between the area of carton pattern and the area of the raw material:

$$W(P) = \frac{\text{Area}(P)}{\text{Area}(\text{AABB}(P))}, \tag{1}$$

where Area(P) is the area of the pattern P and AABB(P) is its axis-aligned bounding box, as illustrated in Fig. 7. It is also possible to use the area of the waste as the measurement of material cost. A user may customize this metric to reflect their design intention. It is supported by Cartonist if the metric is quantified as a scalar value. Note that in our current implementation we assume that the carton pattern has a regular layout on the material and an axis-aligned bounding box is used as the raw material for simplicity. It is also possible to use a general bounding box by allowing a non-axis-aligned setup. The ratio between the area of the 2D pattern and the volume of the 3D folded box is another way to define the material efficiency. Because the dimensions of

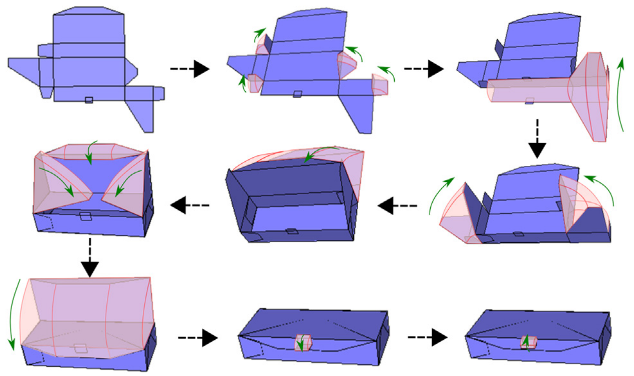


Fig. 8. An example of the folding sequence and the integrated regions used to compute the folding ease. The folding motion is highlighted by green arrows and the integrated regions are highlighted in pink . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the carton box are inputs to our system, this definition reduces to the area of the 2D pattern. We do not use it here because it does not consider the waste material in real cases of fabrication.

Folding Ease. Our Cartonist system also considers the efforts involved in folding the pattern into the carton. To achieve this, Cartonist analyzes the folding sequence of the carton pattern and computes the folding ease from the process. We first define a folding sequence from the 2D carton pattern. The folding sequence is computed from a breadth-first traversal of a tree, with nodes corresponding to the faces on the pattern and edges representing the face–face connectivity [9]. In Cartonist, we follow the similar idea, except that we refine the order of the folding sequence by considering the sealing of the opening mechanisms. We customize the folding sequence for each opening mechanism. For example, we fold the tuck and the gluing tabs before folding their attaching faces. When we traverse the adjacency tree, the folding sequence is adjusted if an opening mechanism is visited. The bottom face of the carton is set as the root of the tree. With the obtained folding sequence, we may use the number of folding steps to measure the folding ease. However, different carton patterns with the same number of folding steps may have different folding efforts. If we are fabricating a large number of cartons, the difference will be notably large. Therefore, we take the shape into consideration to compute the folding ease.

We model the folding ease of the carton pattern as the summation of each folding operation, $E(P) = \sum_i \omega_i E_i(P)$, where $E_i(P)$ is the folding ease of the ith folding operation and ω_i weights the folding operation if it is included in different opening mechanisms. The folding effort is sometimes a subjective value and users may have their own preferences on different opening mechanisms. Therefore, the weights can be specified by the user and we set them to one by default. For each folding operation, we can simulate the motion of each face and use the trajectory of this motion to measure the folding effort. Specifically, if a face F is moving in a folding operation, we model its folding ease as:

$$E_{fi}(f) = -\int_{p\in f} l_i(p)dp, \tag{2}$$

where $l_i(p)$ is the trajectory length of point p on the face f during the ith folding. We sum up the folding eases of all the moving faces for each folding operation to obtain the overall folding ease of the ith folding $E_i(P) = \sum_{f\in P} E_{fi}(f)$. In our implementation, we use numerical integration to compute $E_{fi}(f)$. We use an example to illustrate the folding sequence and the integrated regions for computing the folding ease in Fig. 8. A few panels are folded in each step of the folding sequence and the folding motion of the faces are highlighted by green arrows. We use the pink regions to illustrate the swept volume of the folded motion, which is integrated for measuring the fold ease.

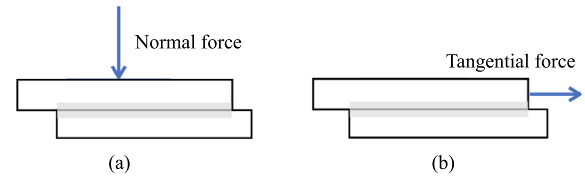


Fig. 9. A larger overlapping area (gray) results in a thicker cardboard to better sustain (a) normal forces and (b) tangential forces, hence enhancing the stability of the carton . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Stability. Our Cartonist system also considers the stability of the carton design. As a container, it is preferable to have the carton well sealed to sustain perturbation and impact during shipment. Because it is costly to model the rich contact, uncertain impacts and material properties to run a precise physical simulation, we adopt a simplified geometric method to approximately measure the stability of a carton. Our insight is that generally larger overlapping area of the folded carton leads to a more stable structure. It is inspired by [23].

There are two types of forces acting on the carton panels. One is the normal force and the other is the tangential force, as illustrated in Fig. 9. Larger overlapping area essentially increases the thickness of the cardboard, leading to more stable design under normal forces. On the other side, a larger overlapping area implies that it needs more effort to resist the friction between the overlapping surfaces, also making the structure more stable under tangential forces. We therefore estimate the stability as the overlapping area of the carton:

$$S(P) = \mathrm{Area}(P) - 2(d_0 \cdot d_1 + d_1 \cdot d_2 + d_2 \cdot d_0), \tag{3}$$

where d_0, d_1, d_2 are the length, width and height of the carton. It computes the difference between the area of the material used for constructing the carton (see Fig. 7) and the total surface area of the constructed carton. Note that we only use an isotropic measurement to approximate the stability of the carton. In the packaging industry, the packed products are usually padded with styrofoams to fill in a carton box. It protects the product and uniformly distributes the stress to the carton. Therefore, we do not consider the anisotropy in the stability and do not model the shape and force direction into this measurement. Based on this simplified mode, we could model the weakest part of the carton as another candidate measurement for the stability. In this case, we compute the overlapping area which corresponds to the face of each opening mechanism and use the minimal one to measure the stability. We could simply replace the measurement function in the system if other measurements are proposed.

3.3. Carton design exploration

With the generated carton patterns and the metrics defined in Section 3.2, we are now ready to build a system to enable exploring the generated designs with different preferences on the design properties. It includes an offline stage to create the shape space of carton patterns and an online stage to explore the designs, as shown in Fig. 4.

In the offline stage, we randomly generate samples of carton patterns using the approach described in Section 3.1 and compute the properties for each sample using the techniques described

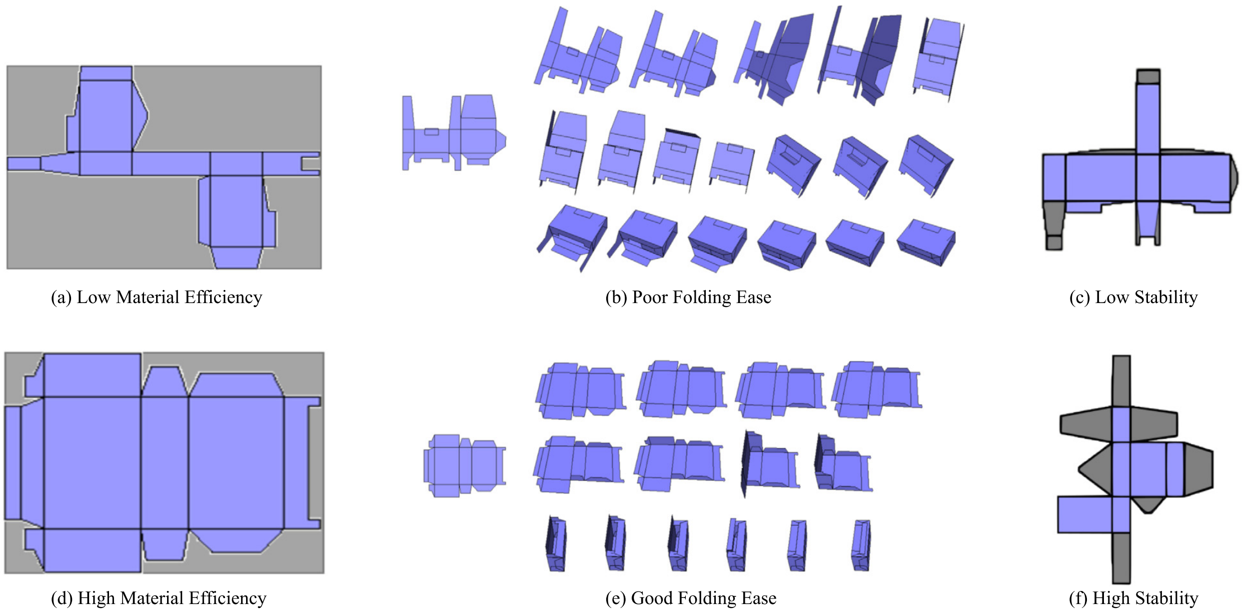


Fig. 10. Different design patterns explored by Cartonist. We show example patterns with (a) low and (d) high material efficiency; (b) poor and (e) folding ease; and (c) low and (f) high stability, where the gray regions correspond to the overlapping areas . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

in Section 3.2. Assume we get the pattern P_i with its properties $\mathbf{x}_i = (W(P_i), E(P_i), S(P_i))$, we embed P_i into the shape space at position $\mathbf{x}_i$ if either of the following conditions holds: the minimum distance to the existing point set $\{\mathbf{x}_j\}_{j<i}$ is larger than a threshold; there is no sample $\{P_j\}_{j<i}$ in the shape space with the same topology and similar geometry as that of P_i. We terminate the offline stage if the number of iterations exceeds a predefined maximum.

In the online stage, a user specifies a target $\mathbf{x}'$ (encoding the desired material efficiency, folding ease and stability) by dragging the corresponding slider bars. Cartonist then searches for the carton pattern P_k whose properties are closest to the desired properties:

$$k = \operatorname*{argmin}_i \|\mathbf{x}_i - \mathbf{x}'\|. \tag{4}$$

To accelerate the search, we consider the samples as 3D points $\mathbf{x}_i$, normalize them and build a KD-tree [24] in the shape space. With this exploration strategy, even if a user specifies infeasible properties, Cartonist still returns the closest choice to fit with the user requirements. In general, users prefer a design with high material efficiency, good folding ease and high stability at the same time. However, the space of carton patterns may not contain samples that are good in all the three properties. Instead, Cartonist suggests a sample whose properties are closest to the desired properties. We also provide the actual properties of the suggested design, and the users are free to adjust the desired properties while seeing the actual properties, in order to trade-off their requirements. We use this sampling method to recommend carton patterns instead of continuous optimization, because the space of valid carton patterns has variations in both topology and geometry. In addition, users may add or customize design properties which are not differentiable with respect to the shape parameters. Therefore, we use this sampling strategy in the exploration. A balanced carton pattern will be recommended if the user does not specify extreme design requirements.

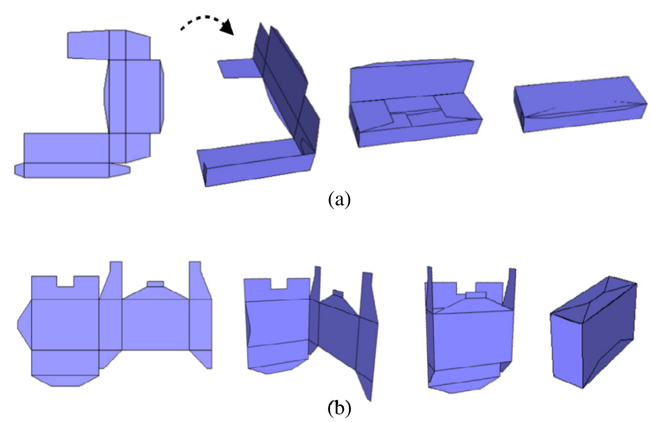


Fig. 11. The folding sequence of two box designs with different dimensions.

4. Experiments

4.1. Results

We implemented our Cartonist system and tested it on a PC equipped with an AMD A8-4500M APU with a single thread. We randomly synthesized 9,000 carton patterns and randomly updated the properties for 10,000 times. The average time for generating a sampled carton pattern is 35 milliseconds and the average time for updating the exploration is about 1 milliseconds. In general, we found it is sufficient to have 10,000 samples in the offline sampling stage and its computation takes around 8 to 10 minutes. For the online exploration, Cartonist can be used to create carton designs in real time.

We show different designs generated from Cartonist in Fig. 10. The left column shows two designs with a lower and higher material efficiency. The middle column shows two results with different measurements of folding ease. In Fig. 10(b, e), the left picture shows the design pattern, and the right picture shows the folding sequences. The carton pattern with good folding ease involves fewer steps and simple panels to be folded in each

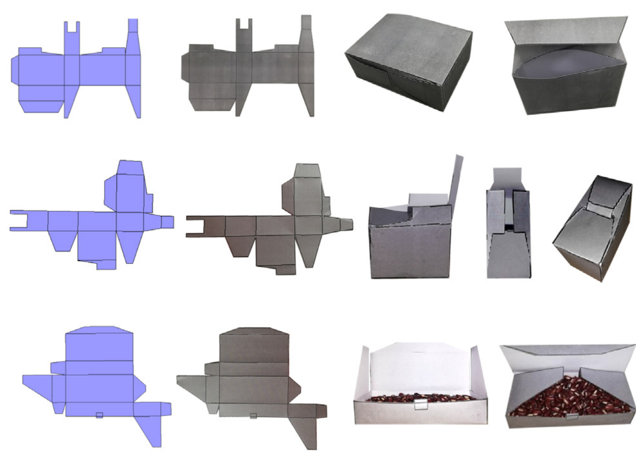


Fig. 12. Three different real-world models created by Cartonist. (a) Designed patterns suggested by Cartonist and (b) the corresponding physical models fabricated in the real world with (c) their folded states.

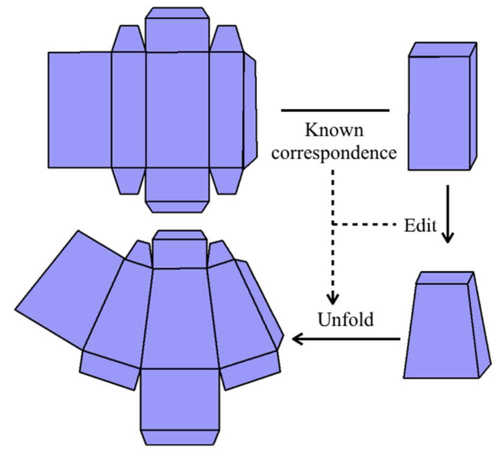


Fig. 13. Our approach can also be extended to create cartons with non-cuboid shapes.

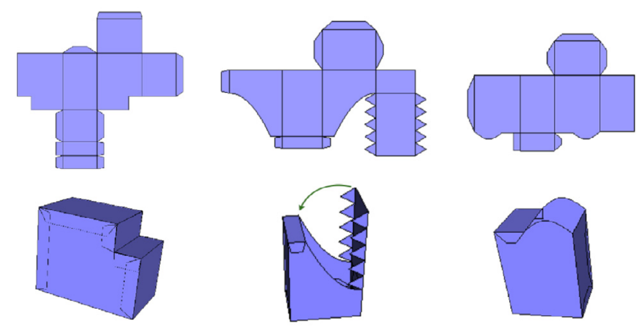


Fig. 14. More irregular cartons with their folding patterns obtained by extending the designed cubic cartons with retargeting or editing operations.

folding step. In the right column, we show two carton patterns, where the design with a larger overlapping area is suggested if the user sets a higher weight for the stability of the design. Our system supports cartons with different dimensions. By specifying different box sizes, more design results are obtained, as shown in Fig. 11, as well as their corresponding folding sequences. Cartonist creates valid patterns that are ready to be fabricated, and we show three real fabricated models in Fig. 12.

Although we focus on the most typical cubic cartons, our Cartonist system can be extended to generate non-cubic cartons. As shown in Fig. 13, we edit the shape of the folded carton in the 3D space to make it non-cubic. By updating the corresponding edges in the 2D space, we can obtain a carton design pattern for the non-cubic folded carton because the correspondence between the folded carton and the 2D pattern is known. In this quick generalization, we only resize the subset of edges which correspond to the edges of the box. This editing may break the constraints introduced in Section 3.1 because the dimensions of the edges on the opening mechanisms are not changed. In this case, we simply project the edges on the opening mechanisms to satisfy the constraints in a greedy way. Because we do not change the topology of the pattern in the editing, the constraints and folding simulation can be reused here. We report failure if the projection fails or the collision detection does not agree with the retargeted pattern. We show three more results of non-cuboid cartons from the generalization in Fig. 14. If one sealing edge of the cubic carton is retargeted to multiple edges, we split the gluing tabs on it to release the tearing stress. An example is shown in Fig. 14(a), where the *L*-shape boundary is enabled by splitting the face to the right side into three panels with more gluing tabs. By adapting the gluing tab into a zigzag shape, cartons with curved boundary is produced in Fig. 14(b). Users are also allowed to delete a few panels to make an open container from the designed carton, as shown in Fig. 14(c). Note that we only propose a possible way to generalize our system for simple non-cuboid cartons. We need to further change the designing method to support more general cartons.

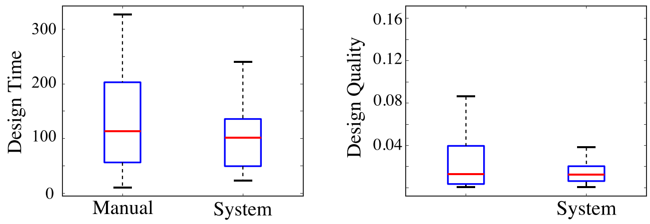
4.2. User study

We conducted a preliminary user study to validate our Cartonist system. The basic hypothesis of this study was that Cartonist is able to help novice users quickly create carton patterns with desired properties. To this end, we recruited 20 participants without any experience in designing carton patterns.

For all participants, we first gave them a 5-minute tutorial of Cartonist. We then had them freely explore the designs using our system until they were satisfied with the result with regard to the desired properties. Because the participants had no experience in designing carton patterns, we did not ask them to design by drawing lines. Instead, we provided them with our system for exploring designs. We refer to this condition as "system mode".

The desired properties $Q_0 = (W_0, E_0, S_0)$ defined in Section 3.2 are randomly generated in this study. We provided both the 2D pattern and 3D folding animation as visualization to the participants. When the participants were satisfied, they confirmed the design by clicking a button. We timed the study and measured the quality of the design as the difference between the properties of the chosen design and the desired properties. After the exploration, we had the participants rate Cartonist using a Likert scale from 0 (poor) to 10 (perfect), to give us their subjective evaluation about whether they like the proposed system.

As a control experiment, we disabled the exploration and only allowed the participants to create and visualize the folding process of random designs. We refer this condition as "manual mode". The goal is still to create a design with the desired properties. Under this condition, a participant clicked a button which triggers our system to generate a random pattern. He could check its folding animation by using the provided user interface. The participant could click the button to generate random designs until he was satisfied with a result. The system then stopped



(a) (b)

Fig. 15. The box plots of (a) the design time (in seconds) and (b) the design quality $\|Q' - Q_0\|$ collected from the manual mode and the system mode. The medians are drawn as red lines. The blue boxes show the first and third quartiles . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

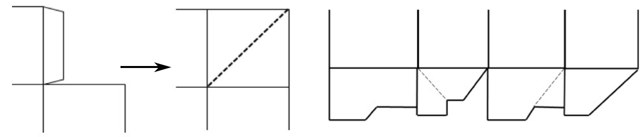
the timing. The wastage, folding ease and stability of the chosen design $Q' = (W', E', S')$ were then stored by the system. After the experiment, we computed the vector distance between Q' and the desired properties Q_0, as a measurement of the quality of the design.

The results are shown as box plots in Fig. 15. Under the manual mode, most of the participants stopped exploration in less than 5 minutes. Under the system mode, the exploration time ranged from 23 seconds to 240 seconds. The design quality $\|Q' - Q_0\|$ of the system mode ranged from 0.001 to 0.038, compared to the manual mode whose design quality ranged from 0.001 to 0.086. It is obvious that the system mode yields better performance over the manual mode in terms of design time and quality.

We also ran paired sample t-tests [25] at the significance level 0.05 to compare the design time and quality between the manual mode and the system mode. In this test, we computed the p-value from the collected data. If the p-value is smaller than 0.05, we can claim that there is less than 5% possibility of no difference between the manual mode and the system mode. In order words, a small p-value (< 0.05) implies a high possibility that the system mode is better than the manual mode. From the collected data, the obtained p-values for the design time and the design quality are 0.03815 (< 0.05) and 0.02825 (< 0.05) respectively, confirming the superior performance (reduced design time and improved design quality) of the system mode over the manual mode at the significance level 0.05.

From the ratings, we found that the users are generally satisfied with the designs suggested by Cartonist and the average score is around 8.00, comparing to the score of 6.15 from the manual mode. The participant who gave the lowest score for the system mode explained that he could not intuitively understand the mapping from the properties to the carton pattern and therefore was not confident about the suggested result. After the study, we showed him the cases in Fig. 10 and he became more clear about how our system worked.

Among the twenty participants, seven of them had experience in using traditional CAD software. They complimented Cartonist on its convenience as it does not require specifying elementary parameters, such as the vertex position and edge length, and they also do not need to manually adjust the design constraints. In contrast, a manually designed pattern might have conflicts on the desired properties, thus complicating the design process. On the other hand, they commented that the suggestions from the design space exploration could serve as a good basis for manual editing, if needed. Incorporating our suggestive design interface into a traditional CAD software could be a good avenue for future extension.



(a) (b)

Fig. 16. Our system can be extended to support more structures in carton design. In (a), a box corner can be created by using a gluing tab or introducing a folding crease. In (b), a mechanism similar to 123-lock is designed with folding creases.

5. Conclusion and future work

In this paper, we present Cartonist, a system that helps users conveniently find valid carton designs under different high-level requirements. We introduce a procedural modeling framework to generate various valid carton designs and propose three high-level metrics to evaluate design properties. With the help of the proposed computational tool, users can easily create carton patterns with desired properties.

Our Cartonist system shows the potential of using a computational design approach to create carton patterns. As part of our future work, we will improve the system with regard to several aspects. First, the rules for synthesizing carton designs can be extended to support more patterns. For example, we could replace the gluing tab with an origami structure as shown in Fig. 16(a) and we could also incorporate more opening mechanisms such as a variant of the 123-lock mechanism as shown in Fig. 16(b). Second, we may incorporate the force direction into the measurement of stability. In this case, how to make a simple user interface to accompany with the elaborated measurement will be an interesting topic, as the force may come from different directions during shipment. Another choice is to further improve the evaluation of the carton patterns by incorporating more sophisticated analysis such as the finite element method [26]. It would be beneficial to study how to simplify the model for a more efficient analysis. Third, we only create simple non-cuboid cartons by retargeting the cartons obtained from our system. A thorough study of the carton patterns for more general folded shapes will make our system more useful. For example, origami might be a good theoretical tool for this problem. Researchers have generalized the origami pattern for complex design even including curved shapes [27]. Although the locking and pasting operations have not been formally studied in origami, we believe it will be interesting to customize origami for general carton design. Finally, as Cartonist has the capacity to create a lot of synthetic data of carton patterns with different properties, it will be interesting to consider the rules of our procedural modeling approach as rules of a game and the properties as reward functions. Based on carton design patterns collected from this work, we will explore if more advanced artificial intelligence techniques can be adopted for carton design.

Acknowledgments

The authors would like to thank anonymous reviewers for their constructive comments. This work was supported by the National Key Technologies R&D Program under Grants No. 2016YFB1001301, the NSFC under Grants No. 61502096 and 61773219, the Zhishan Youth Scholar Program of SEU and Jiangsu Overseas Visiting Scholar Program for University Prominent Young & Middle-aged Teachers and Presidents.

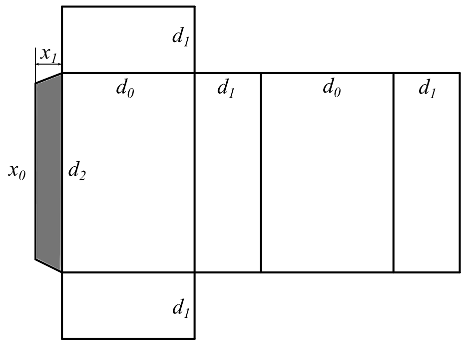


Fig. A.17. A carton trunk (show as white panels) and a gluing tab (shown in gray) . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

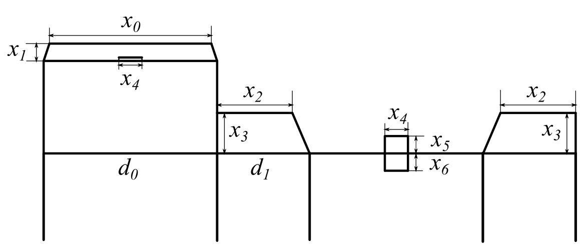


Fig. A.18. Tuck-and-tongue mechanism.

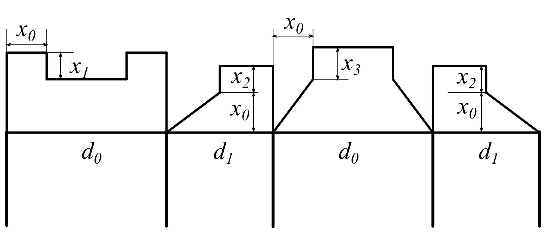


Fig. A.19. 123-Lock mechanism.

Appendix. Carton mechanisms

We list the parameterized mechanisms used in the paper here. The trunk of the carton pattern is parameterized into the width, length and height (d_0, d_1, d_2), as shown in Fig. A.17. The parameters of a gluing tab are also shown in Fig. A.17, where the constraints of the parameters are

$$\begin{cases} 0 \le x_0 \le d_2; \\ 0 \le x_1 \le d_1. \end{cases}$$

The parameters of a tuck-and-tongue mechanism are shown in Fig. A.18, where the constraints of the parameters are

$$\begin{cases} 0 \le x_0 \le d_0; & 0 \le x_4 \le d_0; \\ 0 \le x_1 \le d_2; & 0 \le x_5 \le d_1; \\ 0 \le x_2 \le d_1; & 0 \le x_6 \le d_2; \\ 0 \le x_3 \le d_0; & x_5 = 0 \,\&\, x_6 = 0, \text{ if } x_4 = 0. \end{cases}$$

The parameters of a tuck-and-tongue mechanism are shown in Fig. A.19, where the constraints of the parameters are

$$\begin{cases} 0 \le x_0 \le d_0/2; & 0 \le x_2 \le d_0/2; \\ 0 \le x_1 \le d_1/2; & 0 \le x_3 \le d_1/2; \\ x_0 + x_2 \le d_0/2. \end{cases}$$

References

[1] Demaine ED, O'Rourke J. Geometric folding algorithms: linkages, origami, polyhedra. Cambridge Univ. Press; 2007.

[2] Tachi T. Simulation of rigid origami. In: Origami 4: fourth international meeting of origami science, mathematics, and education, Vol. 4. 2009.

[3] Zhu L, Igarashi T, Mitani J. Soft folding. Comput Graph Forum 2013;32(7):167–76.

[4] Mullineux G, Matthews J. Constraint-based simulation of carton folding operations. Comput Aided Des 2010;42(3):257–65.

[5] Lu L, Akella S. Folding cartons with fixtures: A motion planning approach. IEEE Trans Robot Autom 2000;16(4):346–56.

[6] Yao W, Cannella F, Dai JS. Automatic folding of cartons using a reconfigurable robotic system. Robot Comput-Integr Manuf 2011;27(3):604–13.

[7] Mullineux G, Feldman J, Matthews J. Using constraints at the conceptual stage of the design of carton erection. Mech Mach Theory 2010;45(12):1897–908.

[8] Nimnual R, Suksakulchai S. Virtual reality for packaging folding practice. In: International conference on control, automation and systems; 2007. p. 1011–4.

[9] Shan S, Ma Y, Tang C, Chen X. Folding cartons: Interactive manipulation of cartons from 2D layouts. Comput Aided Geom Design 2018;62:228–38.

[10] Vitalii V, Reginald D, Dirk C, Dirk V, R. DJ. Generating flat patterns for folded structures: Search space reduction. Packag Technol Sci 2018;31(2):97–110.

[11] Javier F, Stephanie G, Colleen T, Laura B. An affordance-based methodology for package design. Packag Technol Sci 2015;28(2):157–71.

[12] Umetani N, Igarashi T, Mitra NJ. Guided exploration of physically valid shapes for furniture design. ACM Trans Graph 2012;31(4):86:1–86:11.

[13] Umetani N, Kaufman DM, Igarashi T, Grinspun E. Sensitive couture for interactive garment modeling and editing. ACM Trans Graph 2011;30(4):90:1–90:12.

[14] Yang Y-L, Yang Y-J, Pottmann H, Mitra NJ. Shape space exploration of constrained meshes. ACM Trans Graph 2011;30(6):124:1–124:12.

[15] Bao F, Yan D-M, Mitra NJ, Wonka P. Generating and exploring good building layouts. ACM Trans Graph 2013;32(4):122:1–122:10.

[16] Schulz A, Xu J, Zhu B, Zheng C, Grinspun E, Matusik W. Interactive design space exploration and optimization for CAD models. ACM Trans Graph 2017;36(4):157:1–157:14.

[17] Alhashim I, Li H, Xu K, Cao J, Ma R, Zhang H. Topology-varying 3D shape creation via structural blending. ACM Trans Graph 2014;33(4):158:1–158:10.

[18] Li J, Xu K, Chaudhuri S, Yumer E, Zhang H, Guibas L. GRASS: generative recursive autoencoders for shape structures. ACM Trans Graph 2017;36(4):52:1–52:14.

[19] Wang G, Laga H, Xie N, Jia J, Tabia H. The shape space of 3D botanical tree models. ACM Trans Graph 2018;37(1):7:1–7:18.

[20] Turney P. Unfolding the tesseract. J Recreat Math 1984;17(1):1–16.

[21] Twede D, Selke SE. Cartons, crates and corrugated board: handbook of paper and wood packaging technology. DEStech Publications, Inc.; 2004.

[22] Winkler T, Drieseberg J, Alexa M, Hormann K. Multi-scale geometry interpolation. Comput Graph Forum 2010;29(2):309–18.

[23] Day RW. Foundation engineering handbook. McGraw-Hill Education; 2010.

[24] Bentley JL. Multidimensional binary search trees used for associative searching. Commun ACM 1975;18(9):509–17.

[25] Leroy G. Designing user studies in informatics. Springer-Verlag London; 2011.

[26] Stefanou G. The stochastic finite element method: Past, present and future. Comput Methods Appl Mech Engrg 2009;198(9):1031–51.

[27] Kilian M, Flöry S, Chen Z, Mitra NJ, Sheffer A, Pottmann H. Curved folding. ACM Trans Graph 2008;27(3):75:1–9.