

# Folding Cartons: Interactive Manipulation of Cartons from 2D Layouts

Submission number: 1074

## Abstract

Cartons are broadly used in packaging industry and our daily life. While most of traditional packages are axis-aligned cuboids, there is a recent trend of creating paper packages and cartons with more complicated geometry. In this work, we present an interactive tool to manipulate 3D cartons from 2D layouts. Two key components, automatic folding and suggestive editing, make our system significantly easy to use. The automatic folding part automatically computes the 3D coordinates of vertices in cartons based on a series of simple rules. Then the suggestive editing part smartly provides a group of geometric constraints, such as vertex merging, shape symmetry, to generate more regular 3D shapes for users to explore.

## 1 Introduction

Cartons have been widely used in packaging industry to deliver various commodities including food items, daily necessities and electronic components. Instead of very basic packaging shapes like cubes, there exist multiple fantastic cartons to package wedding candies or take away coffee cups. These various designs increase much popularity, not to mention they're environmental friendly due to their recycling and degradability [21].

Cartons are usually designed based on experience and trial-and-error, though recent years there are softwares developed to help designers improve efficiency and productivity, for example, KASEMAKE [2] can help users pick the existing structural designs and feed in required basic size and material, also users can re-import the finished artwork to show the print on the structural designs and fold it into a three-dimensional view in seconds. There still consume much work and time to construct 3D model directly from 2D design layout. Moreover, it's sometimes intractable to fold an irregular box layout to final carton without instructions.

Researchers have studied papercraft problem for more than twenty years, they're primarily concerned with the following: (1) computational algorithm and mathematic analysis for folding origami, as in [11, 16, 31]; (2) problems on folding polygon and unfolding polyhedron [3, 23, 25], and some proves to show the problem complexity of folding to polyhedron [4, 5, 18]; (3) other specific structure folding problems like Kinetogami which comprises multi-primitive and reconfigurable units folded from a single sheet of paper [10] and applications to show the folding behaviour of paper [30, 15, 22, 26]. Although previous works solved plenty problems in paper folding, they have not considered intelligent construction 3D models given 2D paper layouts. In this paper, we will focus on the carton folding problem and introduce shape optimization method to solve it.

In this paper, shape constraints for carton are proposed to optimize 2D design layout into the corresponding 3D realization. Moreover, an interactive design and exploration framework is developed to allow users visualize the structural layout in a 3D view, which can improve the efficiency a lot when designing a complicated paper package. The contribution of this paper includes the following:

(1) An interactive and exploration system is presented to manipulate the shape of 3D carton. It creates the corresponding flat polymesh by initialization and user interaction, and the stereo model of the carton is finally constructed.

(2) We propose shape constraints represented by a set of points to implement shape optimization. Observed from existing cartons, constraints are summarized including edge constraint, coplane constraint and plane constraint to keep the shape of cartons. Besides, computer-aid detection like symmetry detection is brought to give users suggestions.

## 2 Related Work

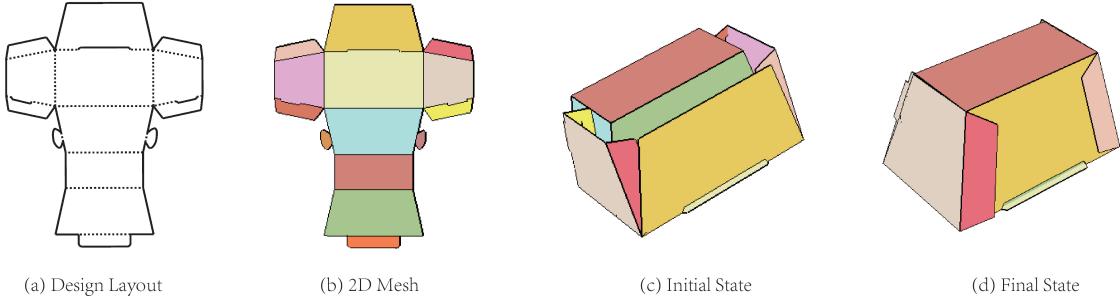
### 2.1 Paper folding problem

Various types of paper crafts have been studied in the field of computation and mathematics. Origami is the Japanese traditional paper art of making different kind of objects by a single sheet of paper, and has been long studied since 1970s [13].

While researching on origami in the field of computational algorithm and geometric analysis, the simulation system is developed to visualize the folding behaviour of a single piece of paper. Thiel provides a virtual origami system including the user interface to model folded paper and show animations of folding process [30]. Kishi et al. allows users create and edit the origami properties over the Web [15]. Nimmual et al. presented an application for package folding practices in a virtual space [22]. Although these applications can model folded paper well, they need given parameters to construct the model, and how to align constraints to structural layout by implementing shape optimization is our main concern.

There are also some methods to solve the problem of carton folding. Song et al. modeled foldable objects as tree like multilink objects and use PRMs(probabilistic roadmap methods [14]) to find a sequence of motions to transform some configuration of a foldable object into another configuration [28]. Mullineux et al. provided a simulation of the carton during erection using a constraint-based approach. Both these work required the target state as a premise, while our work aim to generate the target configuration [21].

Talking about the complexity of folding to polyhedron problem, Lubiw provided an dynamic programming algorithm based on Aleksandrov's theorem to test whether a polygon can be folded into polyhedra which takes  $O(n^2)$  time and



**Figure 1:** Given a 2D layout(a), we first create its 2D mesh(b), and by given a specific angle to each fold edge, we can construct the initial 3D model(c), the final model(d) is finally built through the optimization based on the information acquired from user interaction.

space [18]. O'Rourke examined three open problems on the subject of folding and unfolding [23]. Biedl et al. has studied in polynomial time to solve the question of when is the graph orthogonally convex polyhedra given a graph, edge length and facial angles, also shown that it's NP-hard to decide whether the graph is orthogonally polyhedra or not [5]. Rather than given graph, Biedl et al. proved that if given a net along with the dihedral angle at each crease, we can know whether a net can be folded to a polyhedron in polynomial time, but it becomes NP-hard without the angles even adding constraints on orthogonal polyhedron, which results in more difficulties on more complex input [4]. Compared to our desired result, polyhedron is a set of polygons without overlap, nevertheless, our 3D model contains paste faces that needs to be fixed to another panel. These works above justify our problem being harder to solve causing by even more intricate inputs.

## 2.2 Reconstruction from single line drawings

A line drawing is defined as a 2D projection of object containing its vertices and edges. Line drawings of three-dimensional objects have long been studied, and the main problem is still in object reconstruction given projection on two-dimensional planes. Some researchers treat this task as optimization problem. Marill proposed MSDA(Minimize the Standard Deviation of Angles) principle to emulate the interpretation of line drawings as 3D objects [19]. This new criterion is used by many other researchers later. Leclerc et al. combined MSDA with the deviation from planarity as objective terms [17]. Cao et al. added symmetry measure of the objects to get more complicated results [8]. Some other researchers try to solve this problem from the information theoretic point of view. Marill minimized the description length of objects based on the idea that we usually pick the simplest one from infinite possibilities when we see the line drawing [20]. Shoji et al. implemented the principle of minimizing the entropy of angle distribution between line segments using genetic algorithm [27].

Different from the input above, ours are expanded structural layout of three-dimensional objects in 2D planes, the lackness of 3D topology is main concern in our problem.

## 2.3 Shape optimization

Multiple algorithms have been proposed to enforce shape constraints, and have been used successfully for interactive tools and physical simulation [6, 12]. Bouaziz et al. unified a large variety of geometric constraint into one optimization framework, and provided simple, robust implementation [7]. Poranne et al. described an interactive method to manipulate and optimize polyhedral meshes with constraints with a linear-time algorithm [24]. Tang et al. solved constrained equations by Newton-type method in a fast way, and provided an interactive system to model meshes constrained by equalities and inequalities [29]. Deng et al. developed an interactive tool to explore architectural design with shape constraints, and provided an optimization method to enforce hard constraints and soft constraints at the same time [9].

In our paper, we implement the shape optimization by the method introduced in [7] for its robustness and simplicity.

## 3 Overview

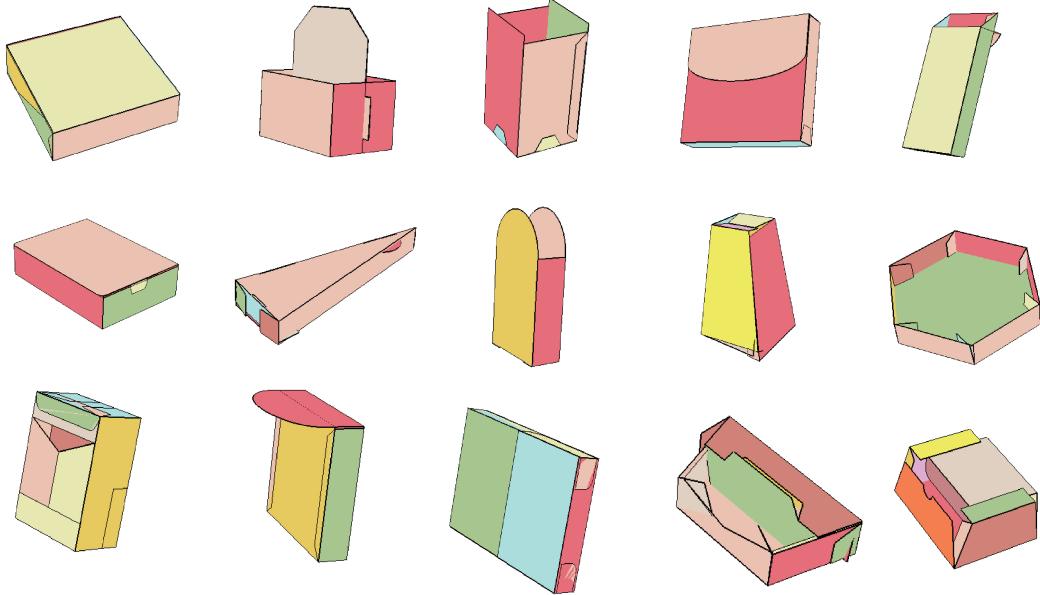
Given a 2D expanded layout of a carton, our system allow users to explore the shape and structure of corresponding models. Figure 1 is our construction algorithm overview. Given a 2D layout Figure 1(a), its 2D mesh Figure 1(b) is created by ignoring the holes in the plane, and the initial 3D model Figure 1(c) is constructed by given a specific angle to each fold edge which will be explained later in section 4.2, then the final model Figure 1(d) is finally built through the optimization based on the information acquired from user interaction.

To formulate our problem, we assume that every plane is rigid which means the interior angle in each plane stays the same, but planes can be bent during folding. Furthermore, planes are connected by hinges at the boundary of patches, and the planar layout has its front and back.

## 4 Algorithm

### 4.1 Definitions

As an input to our initialization, a flat polymesh  $L$  is created from a 2D design layout of a box, then we deform the input polymesh into its 3D realization  $R_i$  according to the predicted angles along each of its fold edges, and through optimization, generate final model  $R_f$ . A polymesh consists of a set of vertices, edges and faces  $M = (V, E, F)$ , the



**Figure 2:** 15 different initialization results. Eight of them need not further refine.

number of vertices  $V$ , edges  $E$  and faces  $F$  vary from one mesh to another. However, a pair of  $(L, R_f)$  as the 2D layout and its corresponding 3D realization share the same topology and therefore they have the same number of vertices, edges and faces. A flat mesh as a 2D layout  $L$  has its  $z$  component of each vertex set to be a constant zero:  $X_z(\mathbf{v}) \equiv 0$  where  $X = (X_x, X_y, X_z)$  is the vertex coordinate, and its normal of each face set as  $(0, 0, 1)^T$ :  $\mathbf{n}(f) \equiv (0, 0, 1)^T$ , where  $f \in F$ .

#### 4.2 Initialization

In this section we explain the reason why using a specific angle to each fold edge, and constructing our initialized model. The basic idea is to interpret the folded state of a box as a series of rotation angles along each edge, where the problem of predicting folded state is turned into a problem of predicting these angles.

Observe from existing data, two common rules are summarized as our basic idea of initialization.

**Plane perpendicularity** The adjacent planes should not be paralleled, and we encourage them to be perpendicular. The term used here is

$$\alpha_{pe} = \sum_{i=1}^n [\sin^{-1}(\mathbf{n}_1 \cdot \mathbf{n}_2)]^2, \quad (1)$$

where  $\hat{\mathbf{n}}_1$  and  $\hat{\mathbf{n}}_2$  denotes all possible combinations of normals of adjacent planes, and  $n$  is the number of such combinations.

**Plane parallelism** It was observed that two planes with same shape are probably paralleled, so we encourage more planes with same shape to be paralleled. The term is

$$\alpha_{pa} = \sum_{i=1}^n [\cos^{-1}(\mathbf{n}_1 \cdot \mathbf{n}_2)]^2, \quad (2)$$

where  $\hat{\mathbf{n}}_1$  and  $\hat{\mathbf{n}}_2$  denotes all possible combinations of normals of disadjacent planes that have same shape, and  $n$  is the number of such combinations.

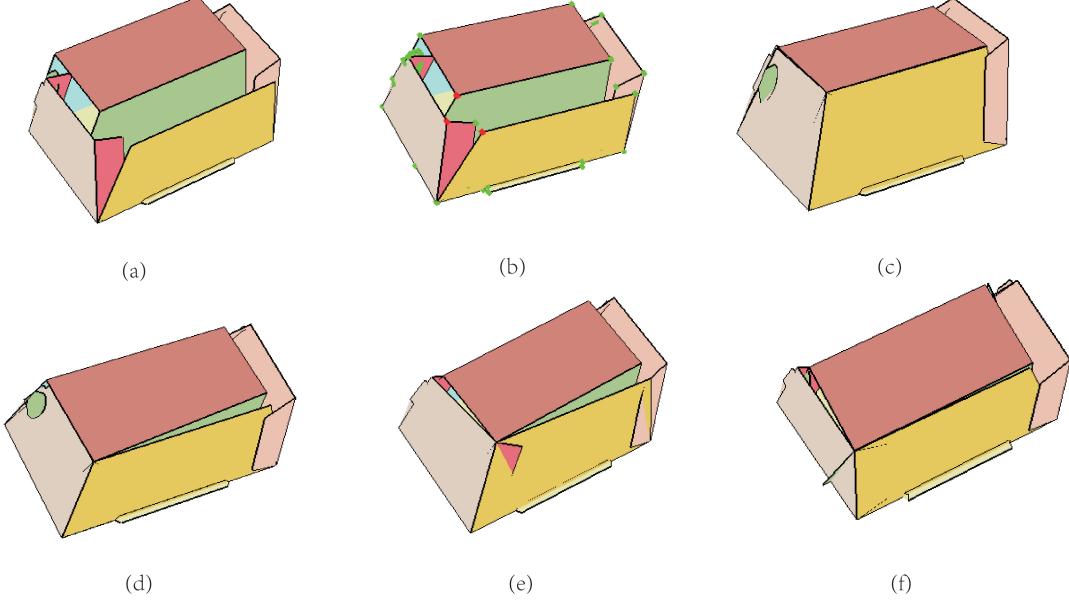
By implementing the CMA-ES(Covariance Matrix Adaptation Evolution Strategy) [1] adding the above two constraints, the initialization result is shown in Figure 2. Note that as the input variables are fold angles of edges,  $\mathbf{n}_1 \cdot \mathbf{n}_2$  is represented by  $\cos\alpha$  where  $\alpha$  is the angle between these two normal vectors.

We can see from the five examples shown in Figure 2, the up three examples can have ideal results and the bottom two results are even not closed at last, which actually caused by the two constraints above lead to a result that each angle of each edge is always  $\pi/2$ . As a consequence, some irregular boxes will present a boxy shape which will be refined later by user interaction.

As for our initialization,  $\pi/2$  is finally set to each angle and this causes more than half of boxes folded correctly in our database.

#### 4.3 Shape Constraint

After initialization, there is still a need to refine the results that have not folded into pleasing results. The main idea is to prescribe the shape constrains by a set of vertices of the polymesh. Moreover, with the extra information acquired from user interaction, we can finally construct the ideal 3D realization compared to the ground truth. As you can see, the coordinate of vertices are chosen as our input instead of fold angles on edges, the reason is that constrains



**Figure 3:** Given an initial state (a), by choosing three points marked in red (b) that need to be relocated together, basic three shape constrains can lead to the optimization result (c). The bottom three are the results optimized lack of edge constrain, coplane constrain and plane constrain separately, compared to (c), the results can not keep the basic shape well.

represented by vertices are simpler and more intuitive than angles, and we can implement the algorithm introduced by Bouaziz et al. easily [7].

We now introduce the constrains used in our construction method:

**Edge Constrain** For each edge  $\{e_j\}_{j=1\dots M}$ , we have its start point  $\mathbf{v}_{js}$  and end point  $\mathbf{v}_{jt}$ , then we have

$$\|\mathbf{v}_{js} - \mathbf{v}_{jt}\|^2 = \|\hat{\mathbf{v}}_{js} - \hat{\mathbf{v}}_{jt}\|^2, \quad (3)$$

to ensure that the length of each edge stays the same.

**Coplane Constrain** For each plane  $\{p_k\}_{k=1\dots P}$  and its normal  $\mathbf{n}_k$ , each line connected by two points  $\mathbf{v}_{ka}, \mathbf{v}_{kb}$  on the plane is perpendicular to the normal.

$$\mathbf{n}_k \cdot (\mathbf{v}_{ka} - \mathbf{v}_{kb}) = 0. \quad (4)$$

**Plane Constrain** For each plane  $\{p_k\}_{k=1\dots P}$ , the length of each line connected by two non-adjacent points  $\mathbf{v}_{ka}, \mathbf{v}_{kb}$  on the plane remains the same, so that the shape of each plane keeps unchanged.

$$\|\mathbf{v}_{ka} - \mathbf{v}_{kb}\|^2 = \|\hat{\mathbf{v}}_{ka} - \hat{\mathbf{v}}_{kb}\|^2. \quad (5)$$

The information acquired from user interaction will add enough constrains to solve the optimization problem, and one of the interaction is to choose the right given suggestion including points needed to be merged together. As for the point information, the constrains can be written like:

**Point Constrain**

$$\mathbf{v}_p - \mathbf{v}_q = 0, \quad (6)$$

if these two points  $\mathbf{v}_p, \mathbf{v}_q$  need to be moved into the same place.

When the above constrains still lead to an ill posed problem, a soft constrain will be introduced:

**Irrelevant Point Constrain** Points  $\{\mathbf{v}_i\}$  which are not in the same plane with  $\mathbf{v}_p$  or  $\mathbf{v}_q$ , should be near the original location, and this constrain just take a light weight  $w$ , which is set 0.001 in our experiment.

$$\mathbf{v}_i - \hat{\mathbf{v}}_i = 0. \quad (7)$$

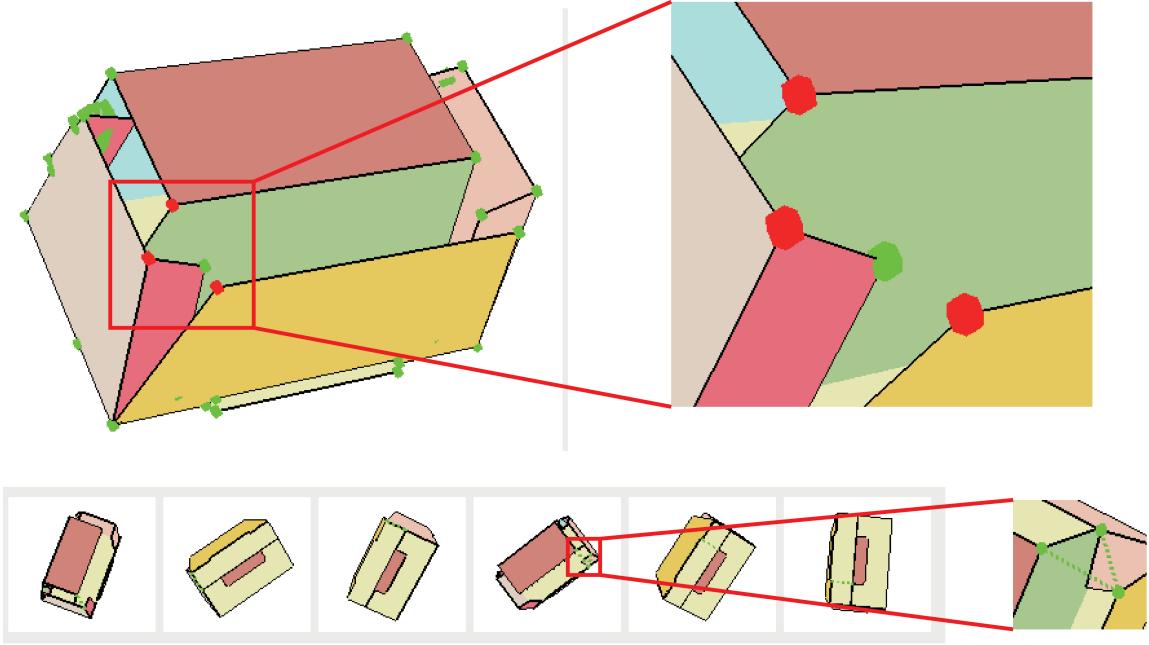
Figure 3 illustrates the necessity of basic three constrains.

#### 4.4 Aided Detection

In order to assist users construct the final carton interactively, we also provide symmetry detection and merging points detection to improve the efficiency of constructing models .

**merging points detection** Consider the initialization results can represent the ideal model partly, the disadjacent vertices that have one edge with same length can be regarded as targets that need to be located in the same place if their Euclidean distance is below a certain threshold.

**symmetry detection** On account of the simplicity of the carton shape, the vertices that have same length set of incident edges can be regarded as symmetric pair. While users select vertices that need to be merged together, symmetry detection can reduce repetitive work by giving users suggestions of symmetric vertices.



**Figure 4:** two operations that the system provides

## 5 User Interaction

After initializing the flat mesh, we need to refine the model to final state through the shape constrain proposed and the information acquired from user interaction. The system provides a set of operations to assist users construct the optimized model, such as selecting points that need to be merged together. Moreover, the system can automatically detect the points that need to be located in one place and provide suggestions to allow users click the right option.

Figure 4 illustrates two operations that the system provides, the first one is selecting points need to be merged by users and the second is select the right option from results by auto-detection.

## 6 Results and Discussion

SS: Any need to list the number of steps we take to generate final model?

The step of shape optimization in our paper can be best demonstrated on the model 5, through initialization, we can have a rough concept of the final model, and by user interaction including selecting vertices merged and faces in the same plane, we can finally generate a pleasing model from a given structural layout.

We show a number of 3D models generated by our system Figure 7, and most of the cases can have an ideal result after initialization because of the cuboid shape as Figure 7 (a). Through interaction, users can folding more complicated cartons to ideal shapes as Figure 7 (b).

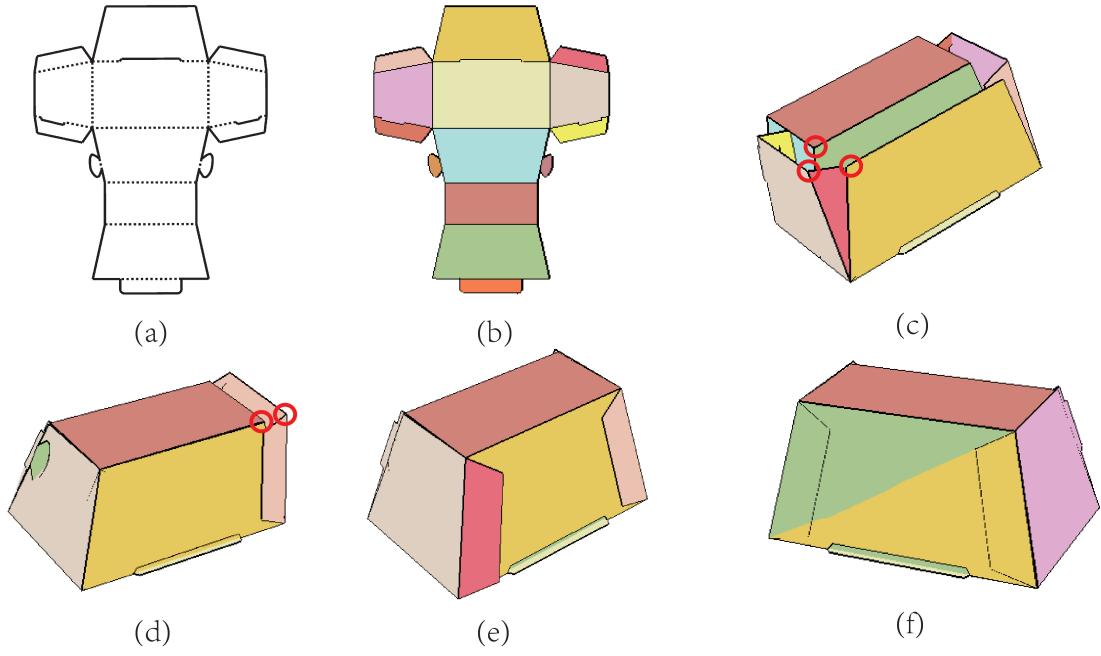
However, the initialization will not generate a pleasing result on some complicated cases, and leads to the difficult interaction to get the final model. As for Figure 6, to have a better corresponding model, users need to take at least ten steps to get to Figure 6(c).

## 7 Conclusion and Future Work

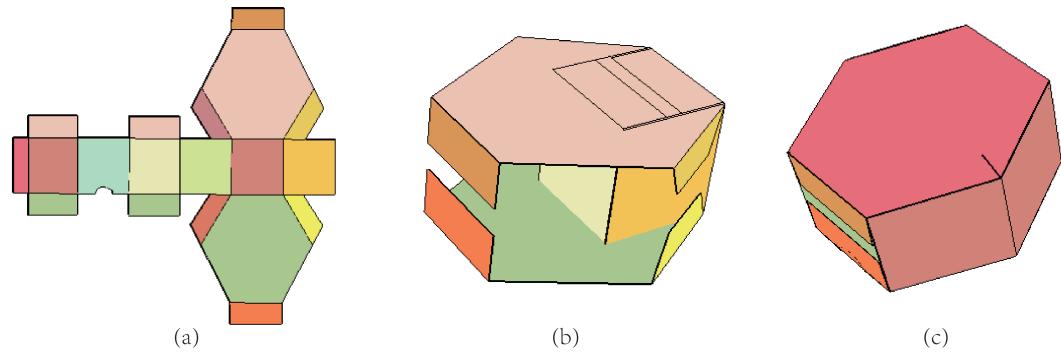
In this paper, we present an interactive modeling system to construct 3D carton realization from 2D expanded structural layout. Simple shape constrains of cartons are also provided to implement optimization on the initial 3D models, which can assist users generate the ideal model efficiently and productively. The direction of future work is adding more operations to our system, includes automatically refining the 2D layouts by 3D models, adding the paste face through stability detection or design the surface pattern of a carton either in 2D or 3D.

## References

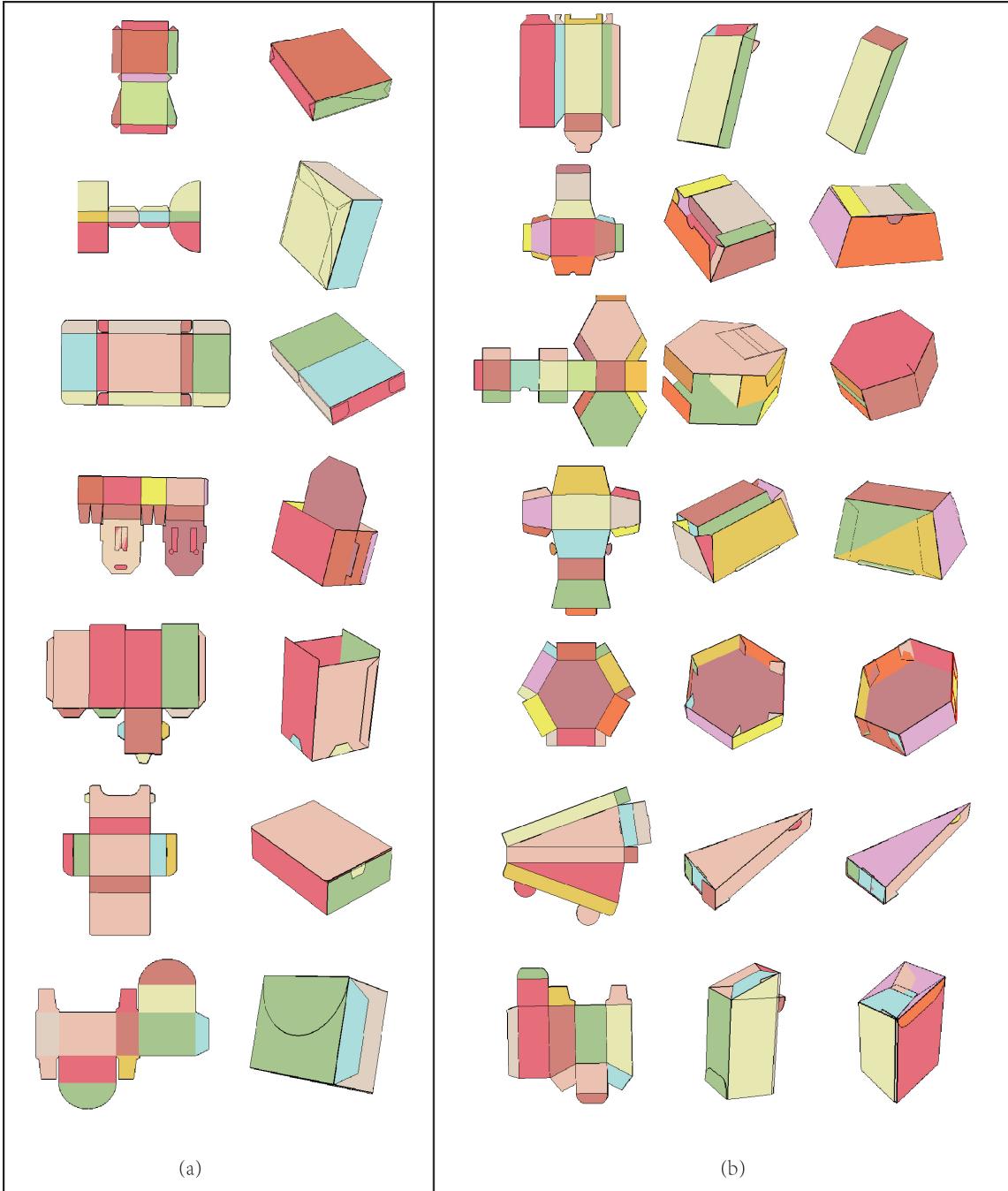
- [1] The cma evolution strategy. <https://www.lri.fr/~hansen/cmaesintro.html>.
- [2] AG/CAD. Kasemake packaging design software. <http://www.agcad.co.uk/software/software.html>.
- [3] M. Bern, E. D. Demaine, D. Eppstein, E. Kuo, A. Mantler, and J. Snoeyink. Ununfoldable polyhedra with convex faces. *Comput. Geom. Theory Appl.*, 24(2):51–62, Feb. 2003.
- [4] 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.
- [5] T. C. Biedl and B. Gensler. When can a graph form an orthogonal polyhedron? pages 100–102, 2004.
- [6] 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*, SGP '06, pages 11–20, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.



**Figure 5:** Structural layout (a) is used to generate flat polymesh (b), and then initialize to a rough model (c), after merge the three vertices in red circles (c) and the two vertices circled red (d), the model is almost closed with two paste faces in red and pink outside the carton (e), finally through selecting these faces are in the same plane of the yellow face as surface, we can have the final model (f).



**Figure 6:** Given a layout as (a), our system can generate an initialized result as (b), and through at least ten steps including select vertices need to be the same place and faces need to be coplane , users can get the final model as (c).



**Figure 7:** More results, part of cartons can be initialized without refining (a), and with interaction, users can manipulate more complicated cartons (b). The first and second column in (a) and (b) are the flat mesh and initialization result of cartons, and the last column in (b) shows the model after interaction.

- [7] 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.
- [8] L. Cao, J. Liu, and X. Tang. 3d object reconstruction from a single 2d line drawing without hidden lines. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1 - Volume 01*, ICCV '05, pages 272–277, Washington, DC, USA, 2005. IEEE Computer Society.
- [9] 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.
- [10] W. Gao, K. Ramani, R. J. Cipra, and T. Siegmund. Kinetogami: A reconfigurable, combinatorial, and printable sheet folding. *Journal of Mechanical Design*, 135(11):111009, 2013.
- [11] T. Ida, H. Takahashi, M. Marin, and F. Ghourabi. Modeling origami for computational construction and beyond. In *Proceedings of the 2007 International Conference on Computational Science and Its Applications - Volume Part II*, ICCSA'07, pages 653–665, Berlin, Heidelberg, 2007. Springer-Verlag.
- [12] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 1134–1141, New York, NY, USA, 2005. ACM.
- [13] T. Kanade. A theory of origami world. *Artificial Intelligence*, 13(3):279 – 311, 1980.
- [14] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. Technical report, Stanford, CA, USA, 1994.
- [15] N. Kishi and Y. Fujii. Origami, folding paper over the web. In *Proceedings of the Third Asian Pacific Computer and Human Interaction*, APCHI '98, pages 337–, Washington, DC, USA, 1998. IEEE Computer Society.
- [16] R. J. Lang. A computational algorithm for origami design. In *Proceedings of the Twelfth Annual Symposium on Computational Geometry*, SCG '96, pages 98–105, New York, NY, USA, 1996. ACM.
- [17] 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.
- [18] A. Lubiw. When can a polygon fold to a polytope? *Dept.comput.sci.smith College*, 1996.
- [19] T. Marill. Emulating the human interpretation of line-drawings as three-dimensional objects. *Int. J. Comput. Vision*, 6(2):147–161, June 1991.
- [20] T. Marill. Why do we see three-dimensional objects? 1992.
- [21] G. Mullineux and J. Matthews. Constraint-based simulation of carton folding operations. *Comput. Aided Des.*, 42(3):257–265, Mar. 2010.
- [22] R. Nimnual and S. Suksakulchai. Virtual reality for packaging folding practice. In *International Conference on Control, Automation and Systems*, pages 1011–1014, 2007.
- [23] J. O'Rourke. Folding and unfolding in computational geometry. In *Revised Papers from the Japanese Conference on Discrete and Computational Geometry*, JCDCG '98, pages 258–266, London, UK, UK, 2000. Springer-Verlag.
- [24] R. Poranne, E. Ovreiù, and C. Gotsman. Interactive planarization and optimization of 3d meshes. *Computer Graphics Forum*, 32(1):152–163, 2013.
- [25] J. O. Rourke. Unfolding polyhedra. *Plus Magazine*, 2008.
- [26] H. Shimanuki, J. Kato, and T. Watanabe. Construction of 3-d paper-made objects from crease patterns. In *Iapr Conference on Machine Vision Applications*, pages 35–38, 2009.
- [27] K. Shoji, K. Kato, and F. Toyama. 3-d interpretation of single line drawings based on entropy minimization principle. 2, 2001.
- [28] G. Song and N. M. Amato. A motion planning approach to folding: From paper craft to protein folding. Technical report, College Station, TX, USA, 2000.
- [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] Z. Xi and J.-M. Lien. Folding rigid origami with closure constraints. In *International Design and Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE)*, Buffalo, NY, Aug. 2014. ASME.