

Folding Cartons:

Submission number: XXX

Abstract

This paper provides an interactive system to construct three-dimensional carton model from two-dimensional design layouts, and simultaneously show these two views to assist users design box layouts more efficiently and productively. We also propose simple shape constraints related to edges and planes to implement the optimized construction along with the acquired information from interaction. To make the interaction simpler, we also provide symmetry detection and automatic detection of the vertices that need to be merged together, and allow users choose the right option.

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 [26].

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 consumes 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 [10, 16, 36]; (2) problems on folding polygon and unfolding polyhedron [3, 28, 29], and some proves to show the problem complexity of folding to polyhedron [4, 5, 22]; (3) other specific structure folding problems, such as origami architectures [17, 20, 21, 30], paper sliceforms [18, 25, 31], Kinetogami which comprises multi-primitive and reconfigurable units folded from a single sheet of paper [9]; (4) applications to show the folding behaviour of paper [12, 15, 27, 32]. 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, we try to construct 3D realization from 2D structural box design directly by shape optimization based on the information obtained from simple user interaction. Besides, we allow users design the pattern in 2D layout, and visualize it in a 3D view, which can improve the efficiency a lot when designing a complicated texture. The contribution of this paper includes the following:

(1) We build an interactive system to design the 2D layout of cartons. It loads the specific type of layout and create the corresponding flat polymesh, by initialization and user interaction, we can finally construct the correct model of the carton.

(2) We propose simple shape constraints represented by a set of points to implement shape optimization. Observed from existing cartons, we can summarize edge constrain, coplane constrain and plane constrain to keep the shape of cartons. Besides, we also provide symmetry detection to give users suggestions.

2 Related Work

2.1 Reconstruction from single line drawings

Line drawings of three-dimensional objects have long been studied, and the main problem is still in reconstruction given projection on two-dimensional planes. Some researchers treat this task as optimization problem. [23] proposed MSDA (Minimize the Standard Deviation of Angles) principle to emulate the interpretation of line drawings as 3D objects. This new criterion is used by many other researchers later. [19] combined MSDA with the deviation from planarity as objective terms. [8] added symmetry measure of the objects to get more complicated results. Some other researchers try to solve this problem from the information theoretic point of view. [24] 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. [33] implemented the principle of minimizing the entropy of angle distribution between line segments using genetic algorithm.

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.

2.2 Folding to polyhedron

[22] provided a 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 space. [28] examined three open problems on the subject of folding and unfolding. [5] 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. Rather than given graph, [4] proved that if given a net along with the dihedral angle

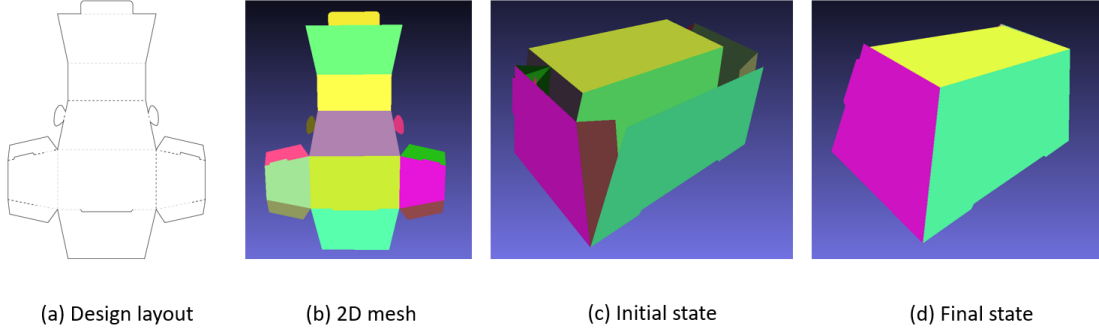


Figure 1: Given a 2D layout(a), we first create its 2D mesh(b), and by given a specific angle to each fold edge, we can construct the initial 3D model(c), the final model(d) is finally built through the optimization based on the information acquired from user interaction.

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.

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 intricater inputs.

2.3 Paper craft

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].

Lately, there have been newly researched automatically generating a type of paper architecture named pop-ups. [21] presented formulation of layouts, sufficient conditions to generate foldable and stable paper architectures, and proposed an automatic algorithm given any 3D models. Plenty papers studied this subject following the idea of Li's and make new improvements [17, 20, 30]. Multiple types of papercraft have been researched as well, such as sliceforms [18, 25, 31], which slots two sets of parallel paper patches together to make a foldable structure approximate to the given object. Cartons are also a kind of paper craft in another meaning.

2.4 Carton Folding

There are also some methods to solve the problem of carton folding. [34] 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. [26] 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.

2.5 Application on folding behaviour

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. [12] provides a virtual origami system including the user interface to model folded paper and show animations of folding process. [15] allows users create and edit the origami properties over the Web. [27] presented an application for package folding practices in a virtual space.

Although these applications can model folded paper well, they need given parameters to construct the model, and we want to simplify these operations by implementing shape optimization.

2.6 Shape optimization

Least square fitting is the common method to enforce shape constraints, and have been used successfully for interactive tools and physical simulation [6, 11]. [7] unified a large variety of geometric constrain into one optimization framework, and provided simple, robust implementation. [35] solved constrained equations by Newton-type method in a fast way, and provided an interactive system to model meshed constrained by equalities and inequalities. In our paper, we implement the shape optimization by the method introduced in [7] for its robustness and simplicity.

3 Overview

We aim to provide a system which help users design a accurate box structural layout and patterns on the surface by showing the three-dimensional view of the folded carton. We can expect a workflow like this: we import the specified type of input and show its 2D structural layout at the same time, then the system will give its initial 3D model and users can interact with the model by choosing the given suggestion to build the final carton model. After that, users can design the texture in 2D view and watch the 3D appearance immediately.

Figure 1 is our construction algorithm overview. Given a 2D layout Figure 1(a), we first create its 2D mesh Figure 1(b), and by given a specific angle to each fold edge which will be explained later in section 4, we can construct the initial 3D model Figure 1(c), 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 the patches, and we assume that the planar layout has its front and back.

4 Initialization

In this section we explain the reason why we use a specific angle to each fold edge, and construct 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.

4.1 Definitions and Notations

As an input to our initialization, we expect a flat polymesh L created from a 2D design layout of a box. As an output of our initialization, we deform the input polymesh into its 3D realization R according to the predicted angles along each of its fold edges. A polymesh consists of a set of vertices, edges and faces $M = (V, E, F)$, the number of vertices, edges and faces vary from one mesh to another. However, a pair of (L, R) 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 Objective Term

Observe from our database includes 60 data pairs (L, R) , we summarize two common rules 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{n}_1 and \hat{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{n}_1 and \hat{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, we can have our initialization result shown in Figure 2. Note that as our input variables are fold angles of edges, we use $\cos\alpha$ to represent $\mathbf{n}_1 \cdot \mathbf{n}_2$ where α 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 we will refine later by user interaction.

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

5 Shape Optimization

After initialization, we still need to refine the results that have not folded into pleasing results. The main idea is to prescribe the shape constraints through deforming using 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 notice, we choose the coordinate of vertices as our input instead of fold angles on edges, the reason is that constraints represented by vertices are more simple and intuitive than angles, and we can implement the algorithm introduced by Bouaziz et al. [7] easily.

5.1 Shape Constrain

We now introduce the constraints we mainly use 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.

Coplanar 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.

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

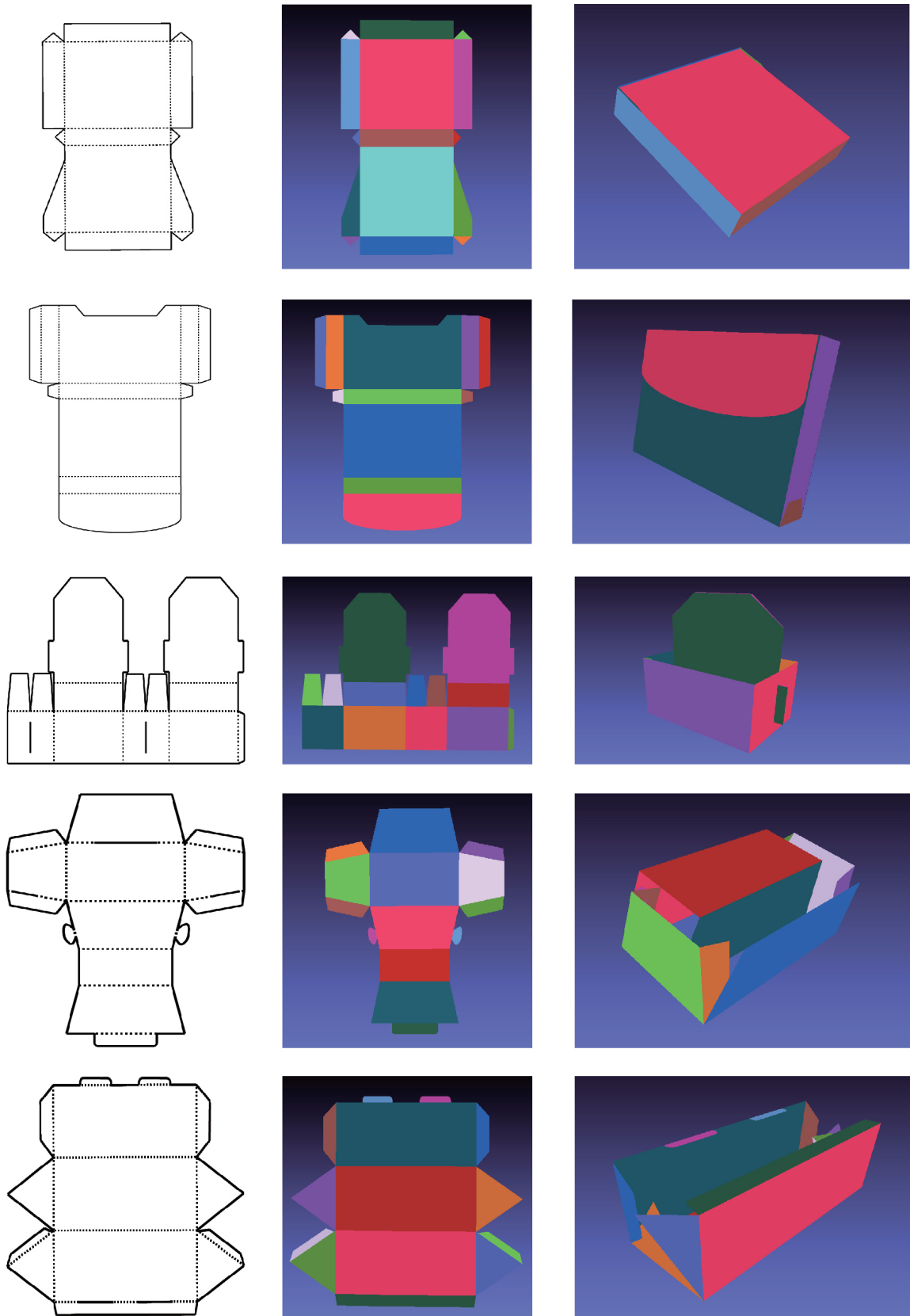


Figure 2: Initialized results of five examples. Each row is an example, and the first column is the 2D design layout, the second column is the polymesh created from layout, the third column is the initialized results optimized with the two constrains.

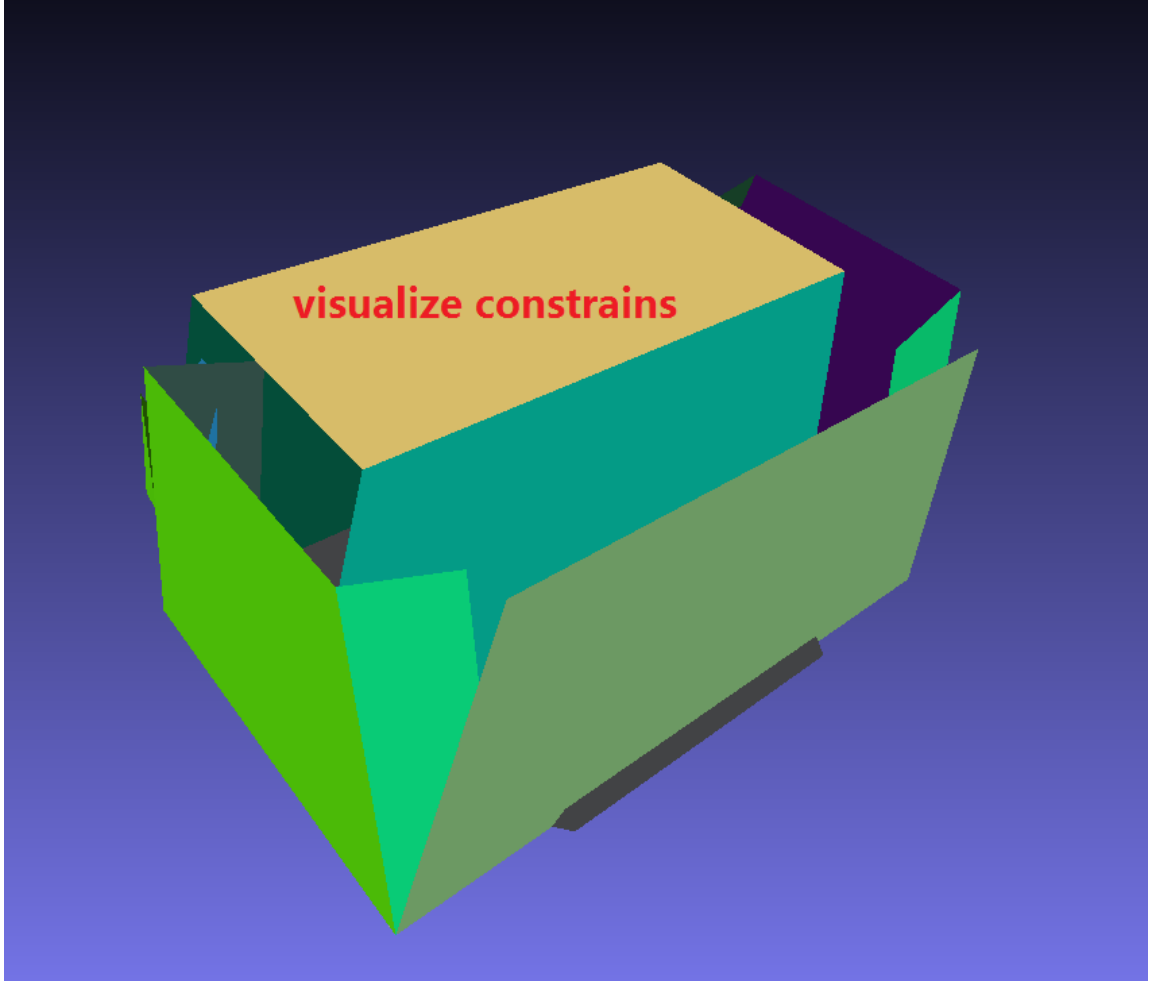


Figure 3

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 help us 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, we can set the constrains like:

Point Constrain

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

if we need to move these two points $\mathbf{v}_p, \mathbf{v}_q$ 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 visually shows the constrains above.

5.2 Aided Detection

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

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

symmetry detection On account of the simplicity of the carton shape, we regard the vertices that have same length set of incident edges as symmetric pair.

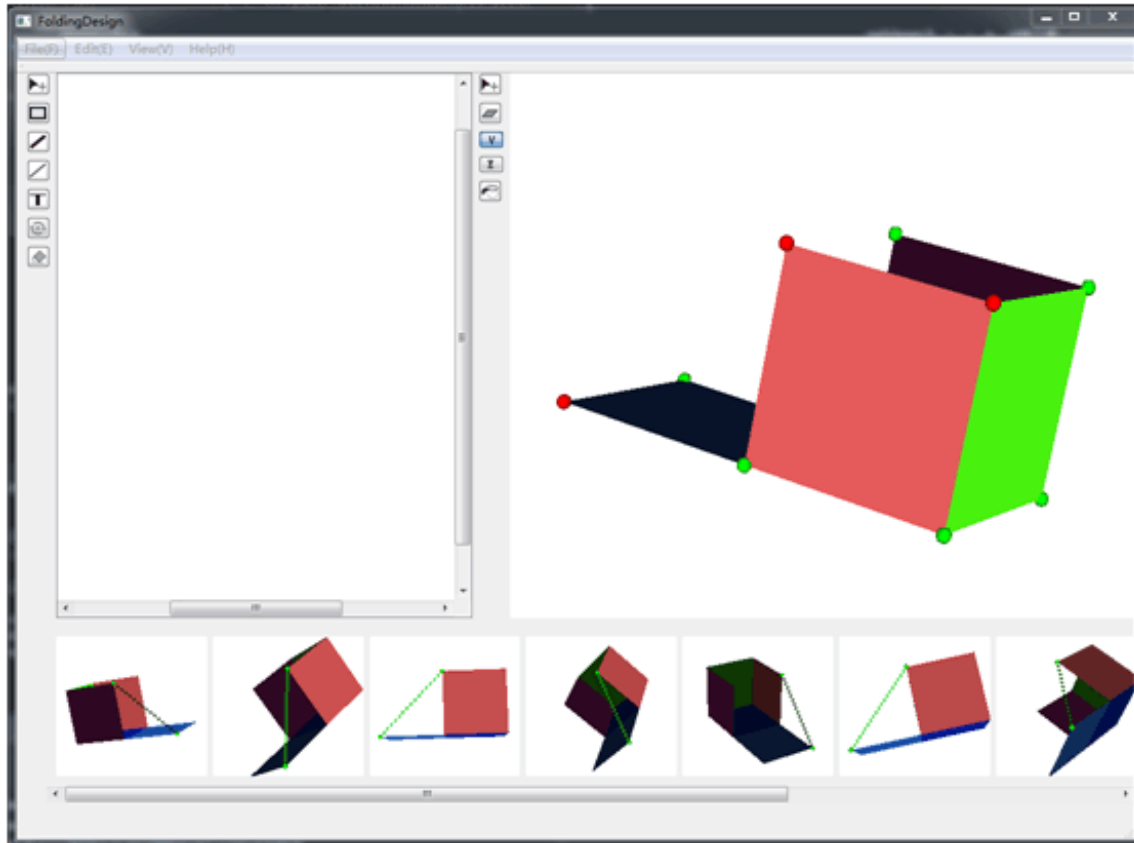


Figure 4

6 User Interaction

After we initialize the flat mesh, we need to refine the model to final state through the shape constrain we propose and the information acquired from user interaction. We provide a set of operations to assist users construct the optimized model, such as selecting points that need to be merged together. Moreover, we 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 shows the interaction we mainly use in our system.

7 Results and Discussion

show every step of results and analysis, show Figure 5, limitation, more results Figure 6

8 Conclusion and Future Work

In this paper, we present an interactive modeling system to construct 3D carton realization from 2D expanded structural layout. We also provide simple shape constrains of cartons 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.

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. Gen̄. 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.
- [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.

vector graph	flat mesh	initialization
merging points detection	symmetry detection	final model

Figure 5

- [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] 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.
- [10] 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.
- [11] 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.
- [12] M. Joanne M. Thiel. Interactive manipulation of virtual folded paper. Master's thesis, UBC, 1998.
- [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] S. N. Le, S.-J. Leow, T.-V. Le-Nguyen, C. Ruiz, and K.-L. Low. Surface and contour-preserving origamic architecture paper pop-ups. *IEEE Transactions on Visualization and Computer Graphics*, 20(2):276–288, Feb. 2014.
- [18] T.-V. Le-Nguyen, K.-L. Low, C. Ruiz Jr., and S. N. Le. Automatic paper sliceform design from 3d solid models. *IEEE Transactions on Visualization and Computer Graphics*, 19(11):1795–1807, Nov. 2013.
- [19] 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.
- [20] X.-Y. Li, T. Ju, Y. Gu, and S.-M. Hu. A geometric study of v-style pop-ups: Theories and algorithms. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, pages 98:1–98:10, New York, NY, USA, 2011. ACM.
- [21] X.-Y. Li, C.-H. Shen, S.-S. Huang, T. Ju, and S.-M. Hu. Popup: Automatic paper architectures from 3d models. In *ACM SIGGRAPH 2010 Papers*, SIGGRAPH '10, pages 111:1–111:9, New York, NY, USA, 2010. ACM.
- [22] A. Lubiw. When can a polygon fold to a polytope? *Dept.comput.sci.smith College*, 1996.
- [23] T. Marill. Emulating the human interpretation of line-drawings as three-dimensional objects. *Int. J. Comput. Vision*, 6(2):147–161, June 1991.
- [24] T. Marill. Why do we see three-dimensional objects? 1992.

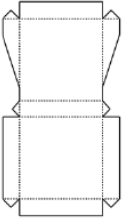
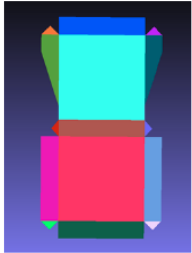
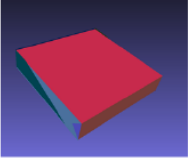
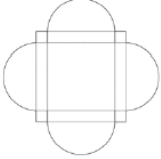
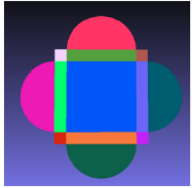
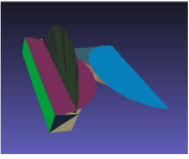
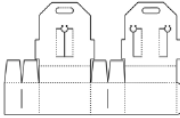

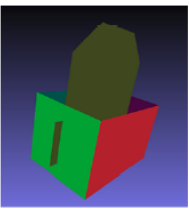
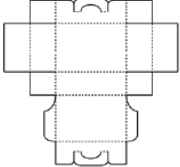

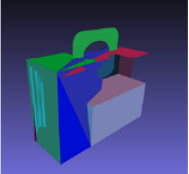
Vector Graph	Flat Mesh	Initialization	Final Model	Ground Truth
				
				
				
				

Figure 6

- [25] J. McCrae, K. Singh, and N. J. Mitra. Slices: A shape-proxy based on planar sections. *ACM Trans. Graph.*, 30(6):168:1–168:12, Dec. 2011.
- [26] G. Mullineux and J. Matthews. Constraint-based simulation of carton folding operations. *Comput. Aided Des.*, 42(3):257–265, Mar. 2010.
- [27] R. Nimmual and S. Suksakulchai. Virtual reality for packaging folding practice. In *International Conference on Control, Automation and Systems*, pages 1011–1014, 2007.
- [28] 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.
- [29] J. O. Rourke. Unfolding polyhedra. *Plus Magazine*, 2008.
- [30] C. R. Ruiz, Jr., S. N. Le, and K.-L. Low. Generating multi-style paper pop-up designs using 3d primitive fitting. In *SIGGRAPH Asia 2013 Technical Briefs*, SA ’13, pages 4:1–4:4, New York, NY, USA, 2013. ACM.
- [31] Y. Schwartzburg and M. Pauly. Fabrication-aware design with intersecting planar pieces. *Computer Graphics Forum (Proceedings of Eurographics 2013)*, 32(2):317–326, 2013.
- [32] 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.
- [33] K. Shoji, K. Kato, and F. Toyama. 3-d interpretation of single line drawings based on entropy minimization principle. 2, 2001.
- [34] 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.
- [35] 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.
- [36] 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.