Cover Letter

We are thankful for the time and efforts by reviewers in reviewing this manuscript and appreciate their insightful and constructive comments. We have carefully revised our paper to address the issues in the review and highlighted all changes in red color. The responds to the reviewers' comments are described below.

Response to comments from reviewer1

1. The technical contribution of this paper is not significant. The whole work seems to be completed while not so solid. For example, only several examples have been shown in the paper. The reviewer expects to see some examples with more structural diversity and complexity.

We have added more examples in Figure 9 and Figure 12. Although the automatically generated examples can also be quickly constructed in other ways, the new added examples in Figure 12 are quite complex to choose angles in a commercial software named STUDIO where users need adjust angles in each creases.

2. For "minimizing the proximity function based on projection operators", it is better to give the proximity function here.

We have added the proximity function for a single point and all vertices in equation 4 and 5.
For a single point $\mathbf{v}$, the proximity function is defined as

$$\Phi(\mathbf{v}) = \sum_{i=1}^{m} \omega_i d_i(\mathbf{v})^2,$$

where ω_i are non-negative weights, and d_i is the distance between the point $\mathbf{v}$ and its projection onto the constraint set, m is the number of constraints. For all the vertices $\mathbf{V} = \{\mathbf{v}_i\}_{i=1\ldots n}$, the shape proximity function is formulated as

$$\Phi(\mathbf{v}) = \sum_{i=1}^{m} \omega_i ||\boldsymbol{N}_i\boldsymbol{V}_i - P_i(\boldsymbol{N}_i\boldsymbol{V}_i)||_2^2,$$

where ω_i are weights and $P_i(\cdot)$ is the projection onto the constraint. The matrix $\boldsymbol{N}_i$ is used to center the vertices at their mean.
The further implementation is not included in the paper and can be find in the corresponding reference.

3. It is better to show the interface of the suggestive system.

The interface has been shown in our video, and we thought it seems a little bit trival if we put the interface figure in the paper. ??????

4. It is better to perform a user study to investigate the effectiveness of the suggestive system.

We conducted a user study to examine the productivity of our system to construct the corresponding 3D model of the planar layout, and the guiding effectiveness of users fabricating the practical mockup. The experiment was separated into two parts. We have added section 7 for user study to explain the procedure and analysis of result.

5. In the related work section, it is better to give more interpretation about the relationship between Paper folding and the problems of Reconstruction from single line drawings and Shape Optimization.

We have added the relationship between our paper folding problem and line drawing problem in Section 2.2, "Although with the same goal that constructing 3D models from 2D layouts including vertices and edges as line drawing problem, the construction process is actually different". Meanwhile, we added the relationship between our paper folding problem and shape optimization in Section 2.3, "The basic idea of our paper is shape optimization by a set of simple algorithms".

6. Paper folding is not an extensively studied problem. To make the paper understandable, it is better to clarify some terms such as Paste face (or give some reference resources).

We have added the function of paste face which ensures the model is stable without manual intervention into the paper, and added an arrow to a paste face in Figure 5 to make sure the audience are clear about the concept of paste face.

7. The visualization of the results need some improvements. Showing only one single-view colourful resulting folded box for each input is not a good visualization. Showing Multi-view and transparent results may be better.

We have added multi-view of generated model in Figure 9 and Figure 12 for visualization. As for the transparent result,??(需要么？)

Response to comments from reviewer2

1. References
#1 Martin Kilian, Simon Flöry, Zhonggui Chen, Niloy J. Mitra, Alla Sheffer, and Helmut Pottmann. 2008. Curved folding. ACM Trans. Graph. 27, 3, Article 75 (August 2008).
-#2 Martin Kilian, Aron Monszpart, and Niloy J. Mitra. 2017. String Actuated Curved Folded Surfaces. ACM Trans. Graph. 36, 3, Article 25 (May 2017).
#3 Solomon, J., Vouga, E., Wardetzky, M. and Grinspun, E. (2012), Flexible Developable Surfaces. Computer Graphics Forum, 31: 1567–1576

Carton folding is closely related to Origami, for which there is an extensive range of existing research. For example:
-#4 Tachi, T., 2009. Simulation of rigid origami. Origami, 4, pp.175-187.
-#5 Tachi, T., 2011. Rigid-foldable thick origami. Origami, 5, pp.253-264.

-#6 Tachi, T., 2010, November. Geometric considerations for the design of rigid origami structures. In Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium (Vol. 12, No. 10, pp. 458-460).

We have added the references and discussion in the Section 2.1 of the revised manuscript. #1 presented an optimization based framework to approximate the given geometric data. #2 provided a subdivision based modeling scheme involving curved paper structure with the folding angle on creases as input. #3 studied the deformation of curved folded surfaces after the folding motion actuated by pulling a network of string. Despite the 2D expanded layout as an input, curved folding problem always has extra information, such as approximate shape, folding angles or external force, while the carton design layout is our only input.

We also mentioned #4 #5 #6 in the folding motion of rigid origami problem and thick origami problem, while thickness is not considered in our problem. In this paper, cartons are folded into 3D model without knowing the prior of crease angles, and can be deformed through optimization while origami folding cannot deal with the deformation.

2. I do not understand in what scenarios would a user need a computer algorithm to help with carton folding. As most carton patterns are designed with a target 3D shapes in mind, the folding instructions should have been generated during the design process. In addition, many of the carton layouts shown in this paper is quite trivial and intuitive, and it seems not necessary to use a computer to figure out how to fold them. The only case I can see where such a technique is helpful is a carton puzzle where the layout becomes rather complicated and there is no folding instructions available. But this seems rather niche. I think the authors should provide a more clear use case for this technique.

We have added the scenario in second paragraph of Introduction, the package design usually need to create a 3D mockup after designing a 2D artwork, and a digital mockup is a great way of showing how the design would appear with lower cost. Based on the 3D mockup, we can design the patterns on the carton and show the appearance in a 3D view. Overall, the generated 3D model is important for designers to design 2D layout.

3. The herustic of approximately 90 degree folding works well for box-like structures, but may not be able to take into account other folding scenarios. For examples, some carton layout involves folding of 180 degrees around an edge to produced a strong enforced face. The heuristic may not be able to capture such case. The authors should provide more discussion on the limitation of the heuristics and maybe present some failure cases.

We have added more discussion on the failure cases in the fifth paragraph of Section 6. Except the case that involves folding of 180 degrees around an edge, the milk carton whose face graph has a loop is also not in our consideration.

Response to comments from reviewer3

1. The writing contains quite a few grammar errors, though fortunately in most cases these do not obstruct understanding of the method. Still, I recommend a thorough editing pass to clean up the *exposition.*

We have revised the grammar errors in the paper, and have checked it for many times.(!!!!!)

2. Is an angle of 90 degrees really the best to assume at the start? For example, a box that rolls up into a prism shape with many sides will need angles much less than 90 degrees, and folding each edge to the full 90 degrees might create a visual mess that is difficult for the user to interact with further. Maybe it makes more sense to start bending each crease up to a maximum of 90 degrees, but stopping if two parts of the box would ever interpenetrate?

Mostly the initialization is good for carton construction, there exist some cases where initialization may seem unreasonable. Adding some simple policy to initialization is a way to improve the initialized result, but it still cannot cover all cases. Take the carton shown in Figure 13(f) as an example, if the initialization stops at an early step, the initialization will be like a pentagonal pyramid which confuses users, and the automatic detection may not work well as the vertices need to be merged are far away.

3. "Face pasting" mentions that in the case that constraints are conflicting, some of them are replaced by soft constraints. Which one? What are the details of the approach?

We have changed the explanation when soft constraint added to our algorithm to “In order to reduce the solution space of the optimization problem”, and in the experiment, the soft constraint is always a part of constraints without replacing any other constraint.

4. In the vertex merging step, nearby vertices are sought, and then checked for compatible adjacent edges. Why not directly look for edges that have equal length and are nearby? This might catch mergable faces too.

The constraint used in our paper is to prescribe the shape of vertices set, and the input of optimization is a set of points, as a result, the automatic detection of vertices is more intuitive for our algorithm. Edges that have equal length and are nearby should keep their endpoints to be merged separately in the implementation, and this operation is still based on vertices in the end.

5. I don't understand the second condition in "Face pasting": won't (v_{ai} - v_c)\cdot n) always equal 0? Why is this check needed?

The second condition in "Face pasting" is defined as

$$|(v_{ai} - v_c^b) \cdot n_b| < \epsilon_d,$$

where v_{ai} is ith vertex in face f_a, and v_c^b is the centroid of face f_b, and n_b is the normal of face f_b, this condition ensures the distance between two faces is under a threshold.

6. Section 5.2 hints at a tool for 3D editing, with the 3D changes mapped back to 2D. What precisely are these editing operations?

The system allows users to select an edge, and move it to the desired place to edit the 3D model, we can see the editing operation from video in 57s.