FoldingBox: From 2D Layout to 3D Model???

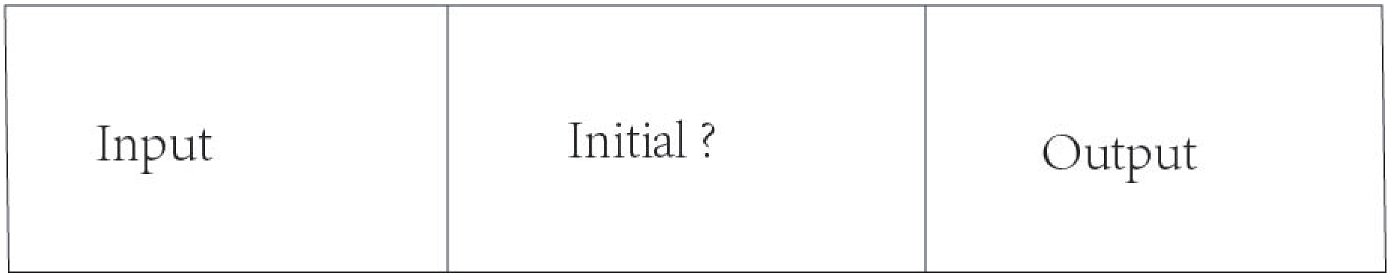

Figure 1: ***Overview.*** *(a) A given input layout (b) XX (c) final 3D model generated by our methods*

Abstract

In this paper, we will propose a new method for generating 3D models given its 2D layout, whose folded angels are learnt(?) by XX optimization.

Keywords: 3D Model?, XX

Concepts:

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. These various designs increase much popularity, not to mention they're environment friendly due to their recycling and degradability.

Cartons are usually designed based on experience and trial-and-error? Depressingly, designers cannot see directly final model of carton when they are drawing its layout. Moreover, for non-experts, it's intractable to fold an irregular layout to final carton without instructions.

Researchers have studied paper folding problem for more than twenty years. It's been verified that given a net, i.e., a polygon and a set of creases, and the dihedral angles at each crease, we can know whether a polyhedron can be obtained in polynomial time. However, if the dihedral angles stay unknown, this problem becomes NP-hard even we simplify the 3D mdoel to orthogonal polyhedron whose dihedral angles are multiples of $\pi/2$ [Biedl et al. 2005].

However, the study above more focused on academic fileds, not on the applications of practical use.

Three-dimensional reconstruction has been wildly studied from different sources, such as point clouds, single images, and line drawings.

SIGGRAPH 2017 Posters, XX, 2017, XX, XX
ISBN: XX
DOI: XX

2 Related Work

Reconsturction 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. [Marill 1991] proposed MSDA(Minimize the Standard Deviation of Angles) principle to emulate the interpretion of line drawings as 3D objects. This new criterion is used by many other researchers later. [Leclerc and Fischler 1992] combined MSDA with the deviation from planarity as objective terms. [Cao et al. 2005] 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. [Marill 1992] 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. [Shoji et al. 2001] implemented the principle of minimizing the entropy of angle distribution between line segments using genetic algorithm. Different from the input above, ours are expanded layout of three-dimensional objects in 2D planes, the lackness of 3D topology is main cencern in our problem.

Folding to polyhedron. [Lubiw 1996] 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 space. [O'Rourke 2000] examined three open problems on the subject of folding and unfolding. [Biedl and Gen 2004] 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, [Biedl et al. 2005] 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 constrains on orthogonal polyhedron, which results in more difficultes on more complex input. Later, [Demaine and O'Rourke 2010] proposed a survey on the folding and unfolding in computational geometry. 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.

Paper craft. Various types of paper crafts have been studied in the field of computing and mathmatics. 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 [Kanade 1980]. Lately, there have been newly researched automaticly generating a type of paper architecture named pop-ups. [Li et al. 2010]

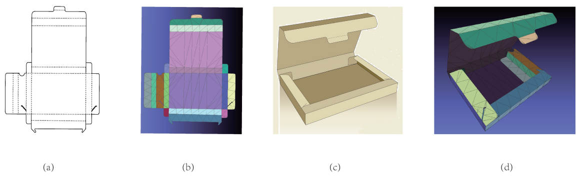

Figure 2: *Given a design layout (a) and its 3D realization (c), we can approximately represent them by triangular mesh as (b) and (d).*

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[Li et al. 2011] [Ruiz et al. 2013] [Le et al. 2014].

Motion planning of paper folding.

3 Problem Formulation

In this section we present the general concept of our learning approach. 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.

3.1 Definitions and Notations

As an input to our method, we expect a 2D designed layout of a box represented as a flat triangular mesh L. As an output of our method, we deform the input triangular mesh into its 3D realization R, according to predicted angles along each of its edges.(An example is shown in Figure 2)

Without loss of generality, learning to predict the folded state of a box is learning an operator $\mathcal{F}$ that maps every 2D layout to its 3D realization:

$$\mathcal{F} : \{L\} \rightarrow \{R\} \tag{1}$$

in which $\{L\}$ is the set of all designed layout of boxes represented by 2D triangular meshes and $\{R\}$ is the set of 3D realization of the same boxes represented by deformed triangular meshes. A triangular mesh consists of a set of vertices, edges and faces $M = (V, E, F)$. The number of vertices, edges and faces varys from one mesh to another. However, a pair of (L, R) as the 2D layout and its correspondent 3D realization share the same topology and therefore has same number of vertices, edges and faces. A flat mesh as a 2D layout from $\{L\}$ has its z component of each vertices set to constant zero: $X_z(v) \equiv 0$ and its normal of each face to $(0, 0, 1)^T$: $N(e) \equiv (0, 0, 1)^T$.

3.2 From Shape Mapping to Functional Mapping

It is difficult to design a model and a learning scheme to learn the operator in (1). As we stressed before, the folded state of a box can be represented by a series of rotation angles along each edge which is why we can learn operator $\bar{\mathcal{F}}$ in (2) instead of $\mathcal{F}$ in (1):

$$\bar{\mathcal{F}} : \{(X_0, X_1, ..., X_n)\} \rightarrow \{\Theta\} \tag{2}$$

in which the $\{(X_0, X_1, ..., X_n)\}$ is a set of feature tensors that is extracted from $\{L\}$ to represent each 2D layout. A feature tensor is composed of n feature functionals and a feature functional has the form of $X_n : E \rightarrow \mathbb{R}$ which is a real value functional that maps each edge of a mesh to a real value. The $\{\Theta\}$ is a set of dihedral angle functionals that maps each edge of a mesh to its angle of folded state. One dihedral angle functional has the form of $\Theta : E \rightarrow [0, 2\pi]$. To avoid ambiguity, we define the dihedral angle to be the one that the face normal pointed to.(need more elaberation with the definition of "positive dihedral angle")

Unfortunately, it is still difficult to learn the operator $\bar{\mathcal{F}}$, since the dimension number of tensors and functionals in $\{(X_0, X_1, ..., X_n)\}$ and $\{\Theta\}$ varies from one pair of (L, R) to another.

To cope with this problem, we approximate each functionals as a linear combination of K basis functionals $\Phi = \{\phi_k\}$ as $X_n \approx \sum \alpha_k \phi_k$ and $\Theta \approx \sum \theta_k \phi_k$. Each pair of (L, R) shares the same Φ, which allows us to further change the problem into learning $\hat{\mathcal{F}}$ as a mapping from the feature coefficient tensors to dihedral angle coefficients:

$$\hat{\mathcal{F}} : \{(\alpha_k\}_0, \{\alpha_k\}_1, ..., \{\alpha_k\}_n)\} \rightarrow \{\{\theta_k\}\} \tag{3}$$

3.3 Choice of Basis Functionals

It is tricky to construct basis functionals $\{\Phi\}$,

3.4 Choice of Features

4 Methods

5 Experiments

We now describe the experimental part of our methods.

5.1 DataBase

To our knowledge, the problem we need to solve is new, thus we have not found any related database. As a result, we built our database following a flow as Figure 3.

For one example as shown in Figure 3(a), we collect the planar layout from designers, then we transfer the design into a flat triangular mesh, and finally interactively fold this triangular mesh into 3D realization.

In the end, we have 51 example pairs where each includes a flat triangular mesh and its corresponding 3D realization.

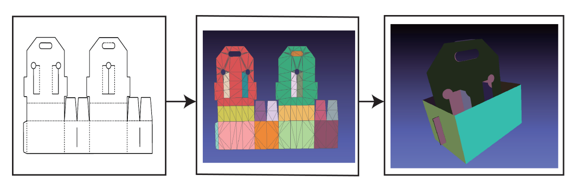

Figure 3: *Take a design layout named box_004 as an example, we collect it from designers, then we convert it to a flat triangular mesh(b), which can finally import to the interface (c) to manully assign angles to each fold edge to get the folded 3D realization.*

5.2 TrainingData

From the database, we extract the ground truth angle as the dihedral angle between front side of two faces.

5.3 Implement Details

Given a flat mesh $\mathcal{M} = (\mathcal{V}, \mathcal{N}, \mathcal{E}, \mathcal{P}, \mathcal{F})$, where

- for each vertex $v_i \in \mathcal{V}, v_z = 0$.
- for each normal $n_i \in \mathcal{N}$, $n_i = (0, 0, 1)$.
- for each edge $\mathbf{e_i} \in \mathcal{E}$, it has a label denotes the line is dash or solid.
- for each plane $p_i \in \mathcal{P}$, it is a minimun rigid polygon.
- for each triangular face $f \in \mathcal{F}$, we know $f_i \in p_j$.

We need to learn angles $\mathcal{A} = \{\alpha_i\}$ of the corresponding edges $\mathcal{E} = \{\mathbf{e_i}\}$, angle α_i means the dihedral angle between front side of two faces sharing edge $\mathbf{e_i}$. There is some constrains on $\mathcal{A}$:

- $\alpha_i = \alpha_j$ if $\mathbf{e_i}$ and $\mathbf{e_j}$ are adjacent and paralleled, also they are in the same plane.??????(how to define)
- $\alpha_i = \pi$ if $\mathbf{e_i}$ is inside the plane.??????(how to define)(does it appear?)

The first constrain means that when two edge are on the same fold edge between two faces, they have the same fold angle. While if edges are inside the plane, the fold angle is always π.

Through $\mathcal{A}$, we can compute the new coordinates of each face which rotates around the edge e_i to angle α_i(need to modify). This just fits the physical problem of folding boxes.

Finally we can get the corresponding 3D realization $\hat{\mathcal{M}} = \{\hat{\mathcal{V}}, \hat{\mathcal{N}}, \mathcal{E}, \mathcal{P}, \mathcal{F}\}$, where $\hat{\mathcal{V}}$ and $\hat{\mathcal{N}}$ is the new computed value of vertices and normals.

6 User Study

7 Conclusion

Acknowledgements

To all.

References

BIEDL, T. C., AND GEN, B. 2004. When can a graph form an orthogonal polyhedron? 100–102.

BIEDL, T., LUBIW, A., AND SUN, J. 2005. When can a net fold to a polyhedron? *Comput. Geom. Theory Appl. 31*, 3 (June), 207–218.

CAO, L., LIU, J., AND TANG, X. 2005. 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*, IEEE Computer Society, Washington, DC, USA, ICCV '05, 272–277.

DAI, J. S., AND JONES, J. R. 1999. Mobility in metamorphic mechanisms of foldable/erectable kinds. *Journal of Mechanical Design 121*, 3, 375–382.

DEMAINE, E. D., AND O'ROURKE, J. 2010. A survey of folding and unfolding in computational geometry. In *Revised Papers from the Japanese Conference on Discrete and Computational Geometry*, 258–266.

KANADE, T. 1980. A theory of origami world. *Artificial Intelligence 13*, 3, 279 – 311.

LE, S. N., LEOW, S.-J., LE-NGUYEN, T.-V., RUIZ, C., AND LOW, K.-L. 2014. Surface and contour-preserving origamic architecture paper pop-ups. *IEEE Transactions on Visualization and Computer Graphics 20*, 2 (Feb.), 276–288.

LECLERC, Y. G., AND FISCHLER, M. A. 1992. An optimization-based approach to the interpretation of single line drawings as 3d wire frames. *International Journal of Computer Vision 9*, 2, 113–136.

LI, X.-Y., SHEN, C.-H., HUANG, S.-S., JU, T., AND HU, S.-M. 2010. Popup: Automatic paper architectures from 3d models. In *ACM SIGGRAPH 2010 Papers*, ACM, New York, NY, USA, SIGGRAPH '10, 111:1–111:9.

LI, X.-Y., JU, T., GU, Y., AND HU, S.-M. 2011. A geometric study of v-style pop-ups: Theories and algorithms. In *ACM SIGGRAPH 2011 Papers*, ACM, New York, NY, USA, SIGGRAPH '11, 98:1–98:10.

LUBIW, A. 1996. When can a polygon fold to a polytope? *Dept.comput.sci.smith College*.

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

MARILL, T. 1992. Why do we see three-dimensional objects?

O'ROURKE, J. 2000. Folding and unfolding in computational geometry. In *Revised Papers from the Japanese Conference on Discrete and Computational Geometry*, Springer-Verlag, London, UK, UK, JCDCG '98, 258–266.

RUIZ, JR., C. R., LE, S. N., AND LOW, K.-L. 2013. Generating multi-style paper pop-up designs using 3d primitive fitting. In *SIGGRAPH Asia 2013 Technical Briefs*, ACM, New York, NY, USA, SA '13, 4:1–4:4.

SHOJI, K., KATO, K., AND TOYAMA, F. 2001. 3-d interpretation of single line drawings based on entropy minimization principle.

WIKIPEDIA, 2016. Chebychevgrblerkutzbach criterion — wikipedia, the free encyclopedia. [Online; accessed 30-September-2016].