Department of Computer Science
George Mason University   Technical Reports

4400 University Drive MS#4A5
Fairfax, VA 22030-4444 USA
http://cs.gmu.edu/   703-993-1530

Compact Folding of Thick Origami via Stacking

Zhonghua Xi, Huangxin Wang, Yue Hao, and Jyh-Ming Lien[*]
Department of Computer Science
George Mason University
Fairfax, Virginia 22032
{zxi, hwang14, yhao3, jmlien}@gmu.edu

In-Suk Choi
High Temp. Energy Materials Research Center
Korea Institute of Science and Technology
Seoul, Republic of Korea
insukchoi@kist.re.kr

Technical Report GMU-CS-TR-2017-1

Abstract

Recent techniques enable folding planer sheets to create complex 3D shapes, however, even a small 3D shape can have large 2D unfoldings. The huge dimension of the flattened structure makes fabrication difficult. In this paper, we propose a novel approach for folding a single *thick strip* into two target shapes: folded 3D shape and stacked shape. The folded shape is an approximation of a complex 3D shape provided by the user. The provided 3D shape may be too large to be fabricated (e.g. 3D-printed) due to limited workspace. Meanwhile, the stacked shape could be the compactest form of the 3D shape which makes its fabrication possible. The compactness of the stacked state also makes packing and transportation easier. The key technical contribution of this work is an efficient method for finding strips for quadrilateral meshes without refinement. We demonstrate our results using both simulation and fabricated models.

1 Introduction

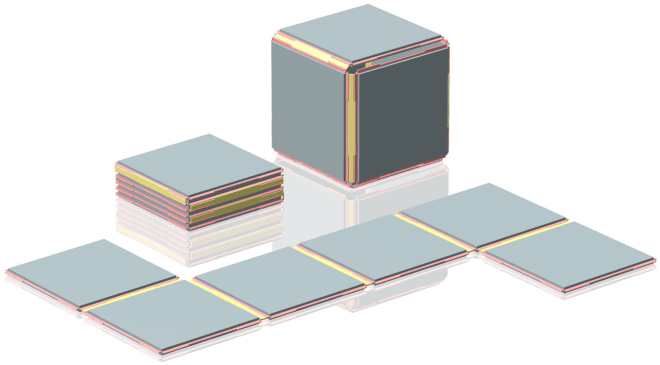


Figure 1: Comparison of the actual size of the folded, unfolded and stacked states of a cube model. The thickness of the panel is 5% of its size.

With the advances in robotics engineering and material science, researchers started to find applications of foldable objects in a wide range of domains including medical devices, self-folding machines, large developable arrays [13, 1, 6, 8]. One main advantage of foldable shapes is the compactness of the folded state as show in Figure 1. This makes carrying of large objects, from maps, umbrellas, chairs, etc., used in our daily life to the solar panels on the satellite, possible and much easier. For example, Zirbel et al. [22] propose a method for designing origami-based deployable arrays with a high deployed-to-stowed ratio which can be used to design solar cells in the spacecraft.

Folding and unfolding also provide opportunities to make 3D shapes that are too large to be fabricated (e.g. 3D-printed) due to limited working volume. Zhou et al. [21] present a method of transforming a 3D shape into a box, however, the volume compression ratio is limited. Nevertheless, most existing methods fabricate foldable objects in 2D space. The area of these unfoldings is often

[*] Address all correspondence to this author.

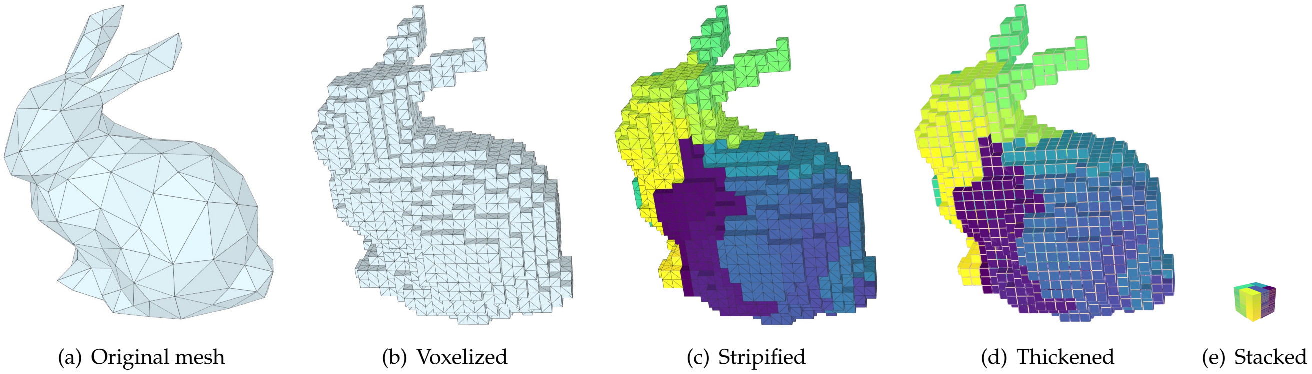

(a) Original mesh (b) Voxelized (c) Stripified (d) Thickened (e) Stacked

Figure 2: Pipeline of our approach.

significantly greater or at least equal to the surface area of the original 3D shape. The huge dimension makes the fabrication of the 2D unfoldings more difficult and sometimes impossible, again, due to limited workspace area. Also, finding unfoldings of complex 3D shapes can be extremely time consuming, e.g. it takes hours to unfold meshes with a few thousands faces [16, 19] and segmentation is required.

In this paper, we propose a novel approach to address both issues by fabricating foldable shapes within limited working space. Our strategy is to find the compactest folded state of the 3D shape. First, we approximate a 3D shape with equal size quadrilateral facets (Figure 2(b)), then we find a Hamiltonian path in the quadrilateral mesh (Figure 2(c)) such that we can fold/stack all faces into one or multiple connected piles (Figure 2(e)). The original model then can be obtained by unfold the piles. Experimental results show that our method can significantly reduce the workspace required to fabricate the models.

2 Background and Related Works

Assembling a polyhedron from one or multiple flat yet foldable/developable components is known as paper crafting which is another practical approach to fabricate complex 3D shapes from flat materials [9, 12, 16, 19]. By cutting along a carefully selected subset of edges of the polyhedron, we could unfold the polyhedron into a planer shape on 2D space without overlapping which is called a net of the polyhedron. Both heuristic methods and evolutionary algorithms are proposed in the literature to find nets of polyhedra [11, 16, 19]. In the rest of this section, we will briefly discuss two research areas closely related to our work.

2.1 Thick Origamis

Mathematical models for folding rigid origami has been developed [14], however, assuming zero thickness material makes it hard to use in practice. Methods for accommodating thick material then were proposed in the literature, including: Axis-shift method [15], this method shifts the rotation axes to either top or bottom of the thick panel depends on the crease type (e.g. mountain or valley); Volume Trimming method [15], this method trims the edge of material to maintain the kinematics to a limited folding angle range; Offset Panel method [5], this method offsets the panels while maintain the rotation axes which can accommodate the full range of motion. Offset Crease method [22], this method widens creases with flexible material and add gaps for folds to accommodate thickness. A detailed comparison of these methods can be found in [10].

2.2 Mesh Stripficiation

The Hamiltonian path/cycle problems are to determine whether there exists a Hamiltonian path/cycle (a path/cycle in the graph that each vertex is visited exactly once) in the given graph which are both NP-complete. The best algorithm so far finds a Hamiltonian cycle in $O(1.657^n)$ for a n-vertex graph and $O(1.251^n)$ for sparse graphs in which every node has a max degree of 3 [7]. Mesh stripification is a special case of the Hamiltonian path problem which has applications in computer graphics for fast rendering, mesh simplification and compression [18]. Taubin [17] finds a Hamiltonian triangulation of a quadrilateral mesh in linear time. It first finds cycles in the dual graph of the mesh and then merge the cycles by flipping the diagonal edges. Although this method is efficient, there is no direct extension to find a single quadrilateral strip. Diaz-Gutierrez and Gopi [4] use the 2-factor partitioning of the dual graph of the quadrilateral mesh to find disjoint cycles and merge those cycles into one. However, this method requires a nontrivial refinement of the mesh to form the Hamiltonian cycle makes it not applicable to stacking problems.

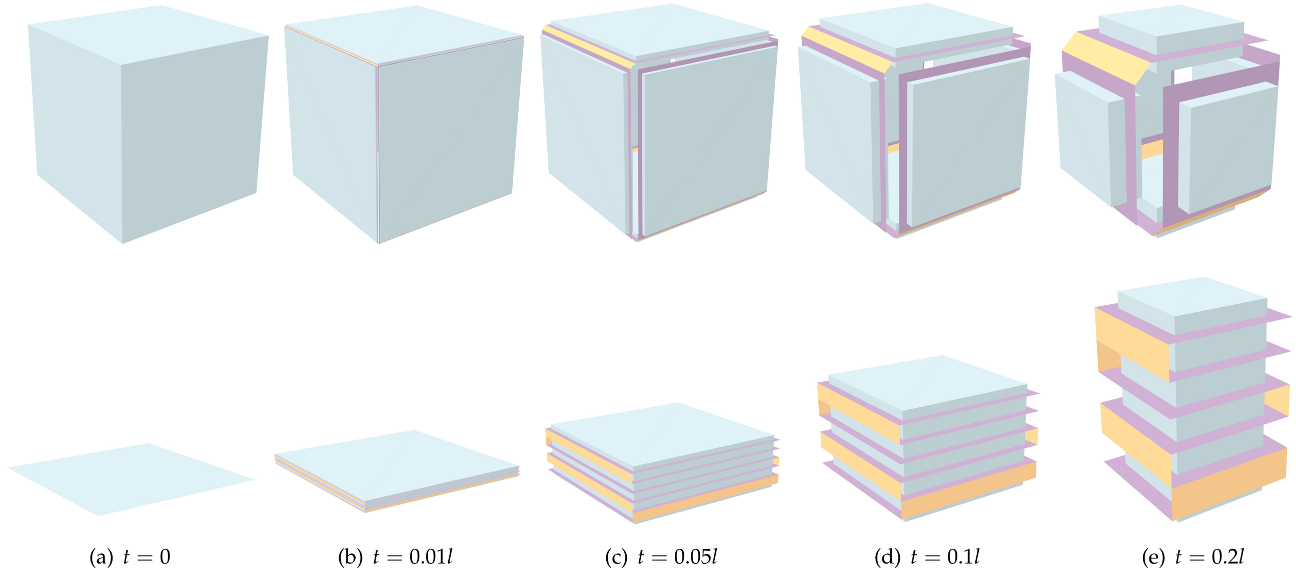


Figure 3: The folded state of a cube model and its corresponding 1-pile stacked state under different thicknesses. l is the original panel size and t is the panel thickness.

3 Stacking Quad Mesh Surface

In this section, we will discuss the stages of the pipeline shown in Figure 2 for stacking the facets of quad mesh. There are four main steps: mesh voxelization, mesh stripification, thickness accommodation and stacking. The main idea of our approach is to approximate the input mesh with equal size square panels. By finding a Hamiltonian path (a strip) of the voxelized mesh, we can stack the strip into a much more compact state. In Figure 2(c), the faces are color-coded to represent the Hamiltonian path. The starting facet is shown in purple, the facet in the middle is shown in green and the ending facet is shown in yellow.

3.1 Creating Quad Mesh via Voxelization

There are many method for creating quadrilateral meshes [?], however, making every quad identical while still being able to approximate the original shape is nontrivial. Voxelization [3] is widely used in compute graphics with applications in visualization, fluid simulation, particle collision, etc. There exists two type of mesh voxelizations: surface voxelization and volume voxelization. In surface voexlization, only the surface region of model will be covered by the voxels while in volume voxelizaiton, voxels are also covered by the inner side of the mesh. We extract the out-most faces from the surface voxelization of the mesh which gives us a perfect mesh for stacking: all faces are identical squares. And each face can fold onto or under one of its four neighboring faces.

3.2 Stripification of Quadrilateral Meshes

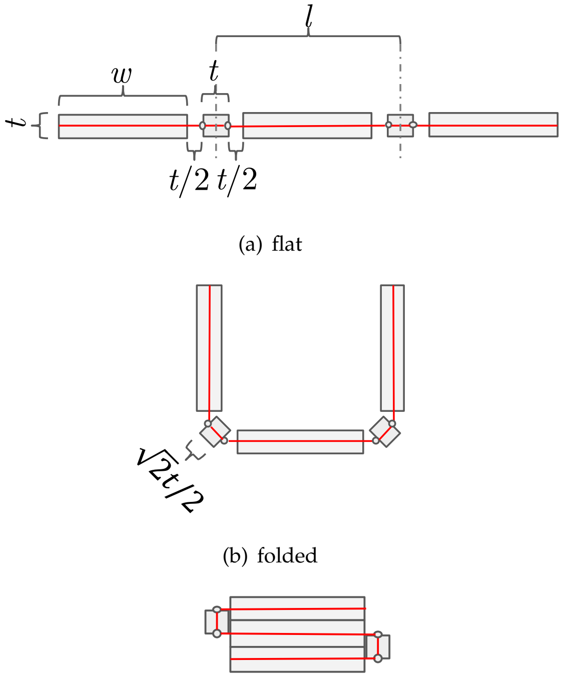
After voxelization, the input mesh is tessellated by identical squares and has exactly four neighbor faces. We are guaranteed to find Hamiltonian paths in the dual graph of the voxelized mesh as a 4-regular graph such that we can cut the mesh and make it stackable.

Traveling Salesman Problem The exponential running time for finding Hamiltonian cycles prohibits its practical usage on quadrilateral meshes of thousands of faces. We convert the Hamiltonian path problem (HPP) into the well known Traveling salesman problem (TSP) which finds a Hamiltonian cycle with minimum cost (the sum of crossed dual edge weights). We set the weight of a pair of nodes to 1 if there is an edge connects them and $+\infty$ otherwise, then we know the optimal solution is n if there are n faces in the original mesh.

There are several reasons for us consider switching to TSP: 1) The solution of TSP can be converted to a set of solutions of HPP. 2) TSP is a well studied problem comparing to HPP, thus many solvers available in the public domain. 3) The known upper bound n helps the solver to cut unnecessary branches thus the solve could find solutions more efficiently. We use public available yet state-of-art TSP solver Concorde [2] to find Hamiltonian paths in the dual graph of the mesh.

3.3 Thickness Accommodation

All previous methods for accommodating thickness have only one target state, the folded state. And a panel only folds in one direction, e.g. it is either a mountain fold or a valley fold. In this paper, we propose a new thickness accommodation method based on the Offset Crease method [22] which enables us fold the panels into two target states, the folded state and the stacked state. We will show that the proposed method guarantees that both target states are self-intersection free. We illustrate our proposed method in Figure 4.



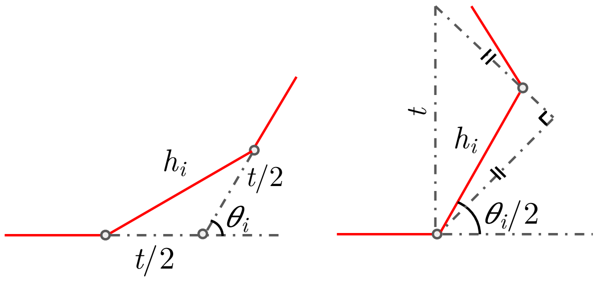
(a) flat

(b) folded

(c) stacked

Figure 4: The proposed method to fold thick panels.

Panels Assuming the original (zero thickness) panel size is $l \times l$, panels of thickness t are trimmed to $(l - 2t) \times (l - 2t) \times t$ to accommodate thickness while enabling them to fold to both directions. Thus the maximum thickness that can be accommodated in our system is $t = 0.5l$.



(a) folding stage: $|\theta_i| \leq \frac{\pi}{2}$

(b) stacking stage: $|\theta_i| > \frac{\pi}{2}$

Figure 5: Hinge length constraints during folding.

Hinges The sliding hinge needs to have a variable length from $\frac{\sqrt{2}}{2}t$ to t during the entire folding process. We categorize the entire folding motion of a hinge into two stages: 1) *folding stage*. As shown in Figure 5(a), $|\theta_i| \leq \frac{\pi}{2}$, we would like to preserve the kinematics as of folding zero thickness material such that we can ensure the final folded state is globally intersection free. 2) *stacking state*. As shown in Figure 5(b), $|\theta_i| > \frac{\pi}{2}$, the goal is to smoothly extend the hinge from $\frac{\sqrt{2}}{2}t$ to t to accommodate the thickness when stacked.

The length of the i-th hinge is derived in Eq. 1,

$$h_i = \begin{cases} cos(\frac{|\theta_i|}{2}) \cdot t, & |\theta_i| \leq \frac{\pi}{2} \\ \frac{\sqrt{2}}{2} \cdot sin(\frac{|\theta_i| - \frac{\pi}{2}}{2}) \cdot t, & |\theta_i| > \frac{\pi}{2} \end{cases} \tag{1}$$

where θ_i is the folding angle of the ideal crease, t is the thickness of the material. When $\theta_i = 0$, the flat state, $h_i = t$; $\theta_i = \pm\frac{\pi}{2}$, the maximum folded state, $h_i = \frac{\sqrt{2}}{2}t$; $\theta_i = \pm\pi$, the stacked state, $h_i = t$. During the entire folding range, we have $\frac{\sqrt{2}}{2}t \leq h_i \leq t$.

Connection We assume panels and hinges are connected by rigid thin material (shown as magenta panels in Figure 3) whose size is $(l - t) \times (l - t) \times t_c$, $t_c \ll t$. The center of connection part is aligned with the center of the panel.

We simulate the folded and stacked states of a cube model with different thicknesses, the results are shown in Figure 3.

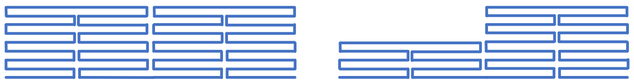
3.4 Stacking

Once we find a Hamiltonian cycle for a mesh, we can break the cycle at arbitrary position to get a strip. For a non-zero thickness panel in the strip, assuming only its neighboring panels can stack with it, one folded onto it and one folded under it. By assigning the folding angles of the panels properly along the path, we can stack all panels of the mesh into one or two piles.

Theorem 1. *A single strip can be stacked into one or two piles.*

Proof. Picking either end of the strip as the base face, a single pile stacking can be achieved by fold each child panel onto its parent panel along the strip. Unfolding the stacking at arbitrary position yields two piles. □

Type of Panels and Piles We say a pile is a *uphill* pile if the heights of its panels along the strip are increasing, otherwise it is a *downhill* pile. The *base panel* of a pile is the panel with height of 0. A penal is the *roof panel* of the pile if it is the highest one in that pile. The roof panel R_{up} of a uphill pile P_{up} can connect with the roof panel R_{down} of a downhill pile P_{down}, while the base panel B_{down} of a downhill pile P_{down} can connect with the base panel B'_{up} of another uphill pile P'_{up}. P_{up} and P_{down} must have the same heights in order to connect at the roof, while P_{down} and P'_{up} can have different heights since they connect at the base. The remaining questions is how to determine the heights for each pile. We discuss two assigning strategies, uniform stacking and non-uniform stacking as shown in Figure 6, to stack the mesh into multiple piles (e.g. 2x2, 3x3), a more compact state, in details in the following.



(a) Uniform stacking

(b) Non-uniform stacking

Figure 6: Front view of two stacking strategies.

Uniform Stacking In uniform stacking, as shown in Figure 6(a), all the piles have the same number of panels. Assuming we always start stacking with a uphill pile then an exception can be made for the last pile if it is a uphill pile. The last uphill pile can have different height, either higher or lower than the rest of the piles. Given a mesh with n panels and the number of piles k, the height h of each pile can be $\lceil n/k \rceil$.

Fold to Stacked State The folding angle of an edge that connects two faces is $\pi - \rho_i$, while ρ_i is the dihedral angle between two faces. Let θ_i and θ'_i denote the folding angle of edge e_i in the original mesh and in the stacked state respectively. Let P_i be the child panel of e_i, that is when e_i rotated, P_i will rotate with e_i.

$$\theta'_i = \begin{cases} 0, & P_i \text{ is a base or roof panel} \\ -\pi, & \text{Height of } P_i \text{ is odd} \\ \pi, & \text{Height of } P_i \text{ is even} \end{cases} \tag{2}$$

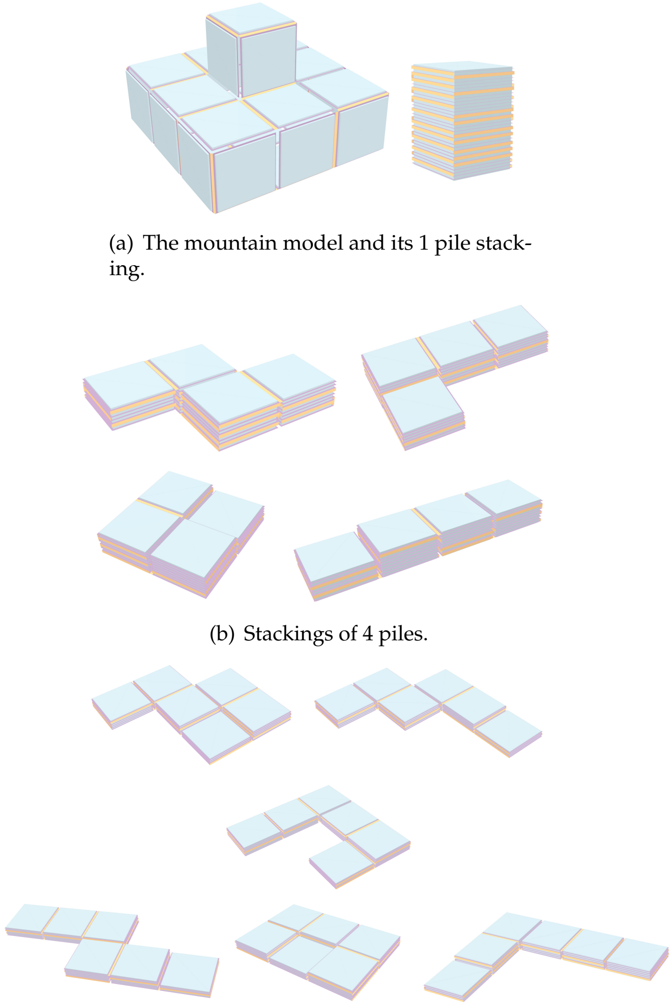
In order to fold the mesh into the stacked state, the angle needed to fold for edge e_i is $\theta'_i - \theta_i$. By assigning the folding angle for each edge based on the type of its child panel, we can fold the mesh into the stacked status in $O(n)$. However, not all stacked states are feasible since some piles might collide with others.

Theorem 2. *A stacking can be validated in $O(n)$ time.*

Proof. First, we align the first face's center to $(0, 0)$, since the coordinate system is discrete, we can check whether a grid has been occupied or not in $O(1)$. We fold one face at a time in the order of the Hamiltonian path. After folding the i-th face f_i, if f_i's center is occupied by another existing pile we know the stacking is invalid. There are $O(n)$ faces in total, thus the validation can be done in $O(n)$. □

By breaking the Hamiltonian cycle at different locations, we have up to $n - 1$ different strips and stacked states. We show a mountain model and its stackings of different number of piles in Figure 7.

Non-uniform Stacking We can relax the same height constraint when there is not enough variation to find a valid folded state via uniform stacking. Each pair of uphill and downhill piles still need to have the same height, while the downhill to uphill pair of piles can have different heights. For simplicity, the height of the later uphill pile is chosen from $\{h, h \pm l\}$, where l can be $1, 2, \cdots, m, m < h$. This gives us $m \cdot (3^{\lfloor k/2 \rfloor} - 1)$ different stacked states for each strip.



(a) The mountain model and its 1 pile stacking.

(b) Stackings of 4 piles.

(c) Stackings of 6 piles.

Figure 7: A mountain model and its representitive stackings of different number of piles.

Compactest Stacking For a given mesh m, we would like to find its compactest stacking state s. The intuition is that the stacking needs to be a box whose width, height and depth should be as close as possible. However, there are many different ways to define the compactness, the optimal stacking may vary depends on the application. In this paper, we adopt a widely used compactness measure: the sum of three dimensions. Under this measure, the compactness ratio (CR) can be defined as:

$$CR = \frac{|W_s| + |D_s| + |H_s|}{|W_m| + |D_m| + |H_m|}, \tag{3}$$

where W, D, H represents the width, depth and height respectively.

$H_s = t\left\lceil \frac{|F|}{W_s D_s} \right\rceil$, Let $W_s = D_s$, then the optimal compactness ratio $CR = \frac{3\sqrt[3]{t|F|}}{|W_m|+|D_m|+|H_m|}$ can be obtained

Table 1: Running time of finding a Hamiltonian path.

Model	# of Quads	Time (s)
Donut	32	0.01
Tower	462	0.59
Table	1964	3.68
Bunny	2206	2.50
Fish	4396	5.41

when $W_s = \sqrt[3]{t|F|}$, where $|F|$ is the number of faces in m and t is the thickness of the panel which has a dimension of 1×1.

We can also measure the volume ratio (VR):

$$VR = \frac{BBox_s}{BBox_m}, \tag{4}$$

which is simply the volume of bounding box of the stacking divided by the volume of the bounding box of the original mesh.

4 Experimental Results

4.1 Experimental Setup

We implemented the proposed method in C++. All data are collected on a Macbook Pro with a 2.5 GHz Intel Core i7 CPU with 16GB Memory running macOS 10.12. We show the models used in the experiments in Figure 8.

4.2 Finding Hamiltonian Paths

We use Concorde TSP solver to find Hamilton paths in the dual graph of the mesh. The found Hamiltonian paths are coded in Figure 8, start with purple and end with yellow with green as the intermediate color. The running time for finding a Hamilton path on different models in Table 1 from which we can see that the running times grow almost linearly to the number of quads in the mesh.

4.3 Compactest Stacking

We show the compactest stackings of the Bunny model shown in Figure 8(d) under different thickness in Figure 9. The dimension of bounding box of the bunny model is $22 \times 19 \times 27 (W \times D \times H)$. The compactness ratio and volume ratio of the optimal stackings of the Bunny model is listed in Table 2, from which we can see that even the thickness of the panel is 30% of its width, the optimal stacked state only takes about 6% of volume of the bounding box of the unfolded model.

4.4 Finding Continous Folding Motions

We use motion planning technique to find continuous folding motion between the stacked state the target state. A discrete domain sampling based planner [20] is employed to find such motion to ensure it is physically realizable. We show the folding motions of the Cube (shown in Figure 3) and Mountain model (shown in Figure 7) from their 1-pile stacking to the unfolded/target shapes in Figure 10 and Figure 11 respectively. The running time for the planner to find a continuous folding path, on average of 30 trails, is 0.05 seconds for the Cube model and 79.09 seconds for the Mountain model.

Table 2: The optimal compactness ratio (CR) and volume ratio (VR) of the stacked Bunny model under different thicknesses t.

t	Stacking ($W \times D \times H$)	CR	VR
0.0005	$1 \times 1 \times 1.103$	0.0456	9.7732×10^{-5}
0.005	$2 \times 2 \times 2.760$	0.0994	9.7820×10^{-4}
0.01	$3 \times 3 \times 2.460$	0.1244	1.9617×10^{-3}
0.1	$6 \times 6 \times 6.200$	0.2675	1.9778×10^{-2}
0.2	$8 \times 8 \times 7.000$	0.3382	3.9695×10^{-2}
0.3	$9 \times 9 \times 8.400$	0.3882	6.0287×10^{-2}

4.5 Physical Models

We show a physical realization of the cube model using Lego to illustrate the proposed idea of two foldable target shapes. The panel size is 3.2in × 3.2in × 1.0in, the size of ideal zero thickness panel will be 3.6in × 3.6in. The net, the folded state and the stacked state of the cube model are shown in Figure 12. We also show two others shapes that can be folded from the cube chain.

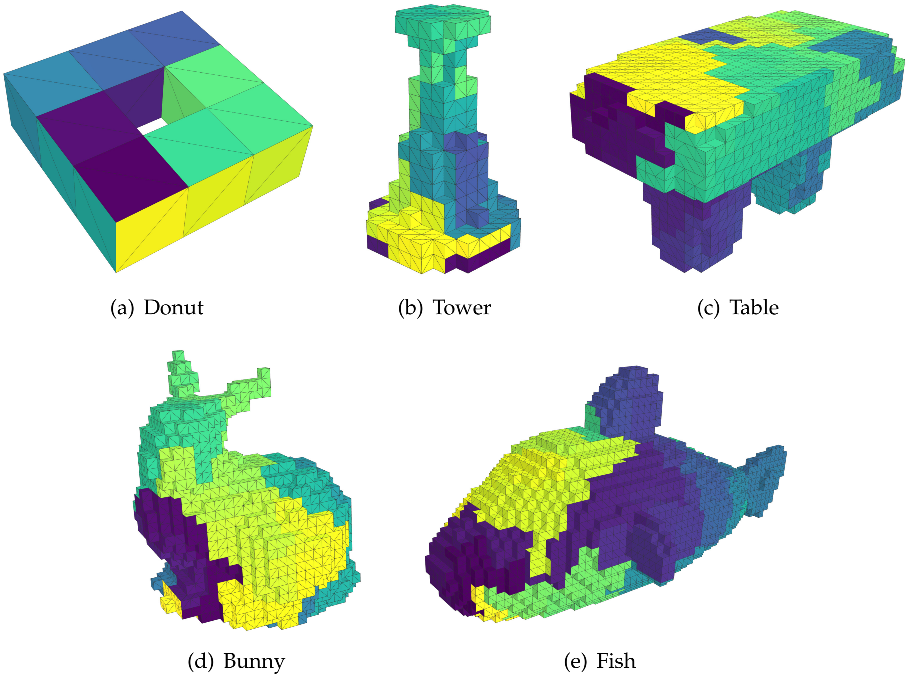
5 Conclusion

In this paper, we propose a novel approach to fold a voxelized mesh into a much more compact form called stacking which enable us to fabricate (e.g. 3D-printing and unfolding) a large 3D model from a much smaller workspace. We show a technique to accommodate the thickness of the material which also enables the folding motion in both directions.

Limitations and Future Works In this work, we did not plan the folding motion for those complex thick chains by assuming there always exists a collision folding path due to its huge degree of freedom (DOF). Meanwhile, the high degree of freedom (DOF) makes the thick chains hard to fold for both humans and themselves as self-folding machines. We are seeking other representations instead of voxelization of the mesh to obtain a better yet stackable approximation of the original model.

References

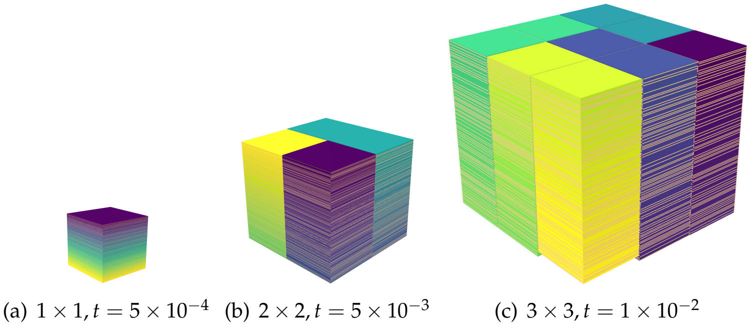
[1] Saad Ahmed, Carlye Lauff, Adrienne Crivaro, Kevin McGough, Robert Sheridan, Mary Frecker,



(a) Donut (b) Tower (c) Table

(d) Bunny (e) Fish

Figure 8: Model used in our experiments. The Hamiltonian path is color coded from purple to yellow via green.



(a) $1 \times 1, t = 5 \times 10^{-4}$ (b) $2 \times 2, t = 5 \times 10^{-3}$ (c) $3 \times 3, t = 1 \times 10^{-2}$

Figure 9: The compactest stacked states of the Bunny model (Figure 8(d)) under different thicknesses.

Paris von Lockette, Zoubeida Ounaies, Timothy Simpson, Jyh-Ming Lien, and Rebecca Strzelec. Multi-field responsive origami structures: Preliminary modeling and experiments. In *Proceedings of the ASME 2013 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, August 2013.

[2] David Applegate, Ribert Bixby, Vasek Chvatal, and William Cook. Concorde tsp solver, 2006.

[3] Daniel Cohen-Or and Arie Kaufman. Fundamentals of surface voxelization. *Graphical models and image processing*, 57(6):453–461, 1995.

[4] Pablo Diaz-Gutierrez and Meenakshisundaram Gopi. Quadrilateral and tetrahedral mesh stripification using 2-factor partitioning of the dual graph. *The Visual Computer*, 21(8-10):689–697, 2005.

[5] Bryce J Edmondson, Robert J Lang, Spencer P Magleby, and Larry L Howell. An offset panel technique for thick rigidily foldable origami. In *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2014.

[6] S Felton, M Tolley, E Demaine, D Rus, and R Wood. A method for building self-folding machines. *Science*, 345(6197):644–646, 2014.

[7] Kazuo Iwama and Takuya Nakashima. An improved exact algorithm for cubic graph tsp. In *International Computing and Combinatorics Conference*, pages 108–117. Springer, 2007.

[8] Dae-Young Lee, Ji-Suk Kim, Jae-Jun Park, Sa-Reum Kim, and Kyu-Jin Cho. Fabrication of origami

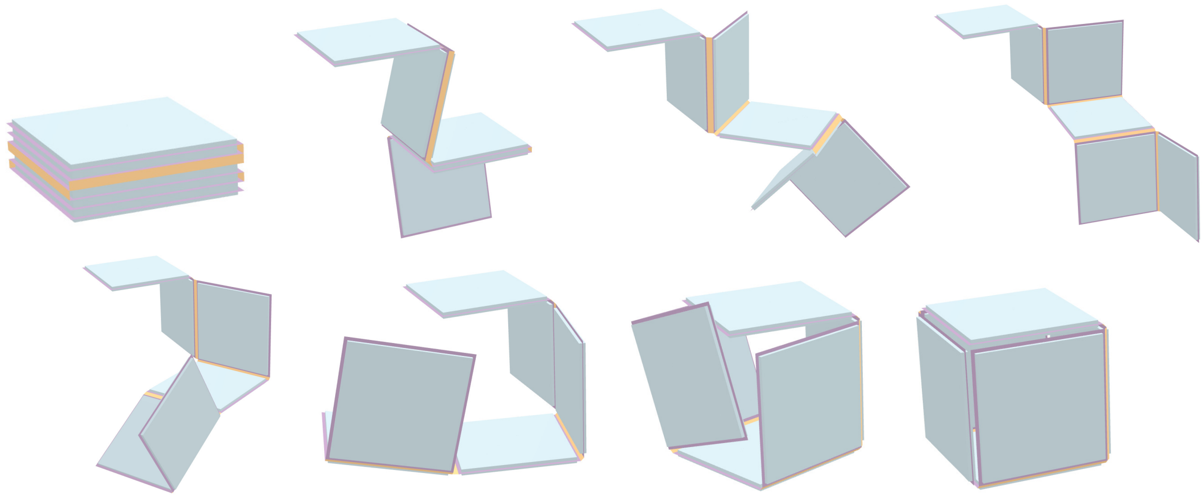


Figure 10: The continuous unfolding motion of the Cube model from its stacked shape to target shape. The folding motion can be best visualized using our web-based interactive folder at https://goo.gl/cmQjN9.

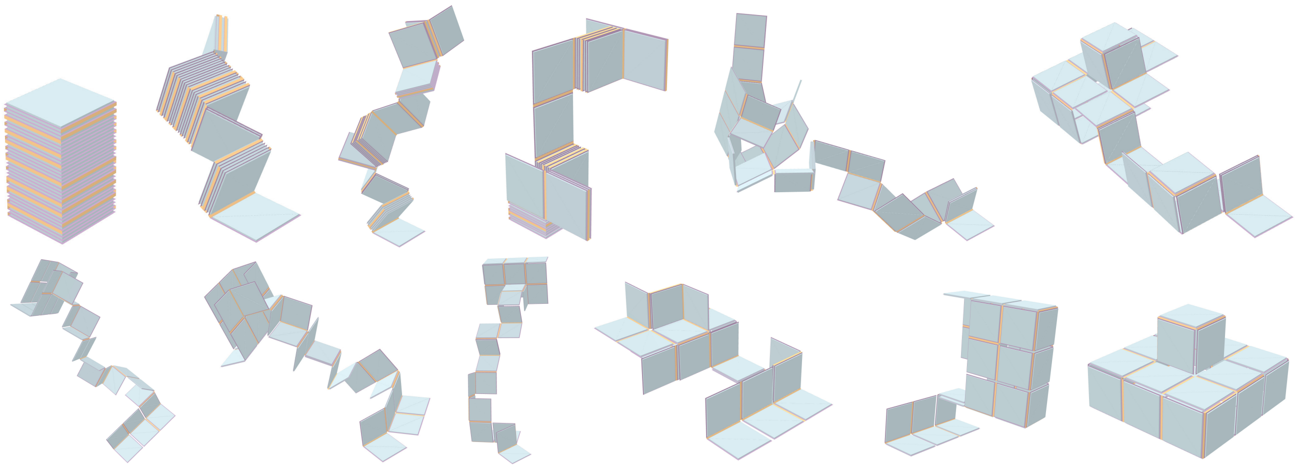


Figure 11: The continuous unfolding motion of the Mountain model from its stacked shape to target shape. The folding motion can be best visualized using our web-based interactive folder at https://goo.gl/BDSWbd.

wheel using pattern embedded fabric and its application to a deformable mobile robot. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2565–2565. IEEE, 2014.

[9] Jun Mitani and Hiromasa Suzuki. Making papercraft toys from meshes using strip-based approximate unfolding. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 259–263. ACM, 2004.

[10] Michael R Morgan, Robert J Lang, Spencer P Magleby, and Larry L Howell. Towards developing product applications of thick origami using the offset panel technique. *Mechanical Sciences*, 7(1):69, 2016.

[11] Wolfram Schlickenrieder. Nets of polyhedra. Master's thesis, Technische Universität Berlin, 1997.

[12] Idan Shatz, Ayellet Tal, and George Leifman. Paper craft models from meshes. *The Visual Computer*, 22(9-11):825–834, 2006.

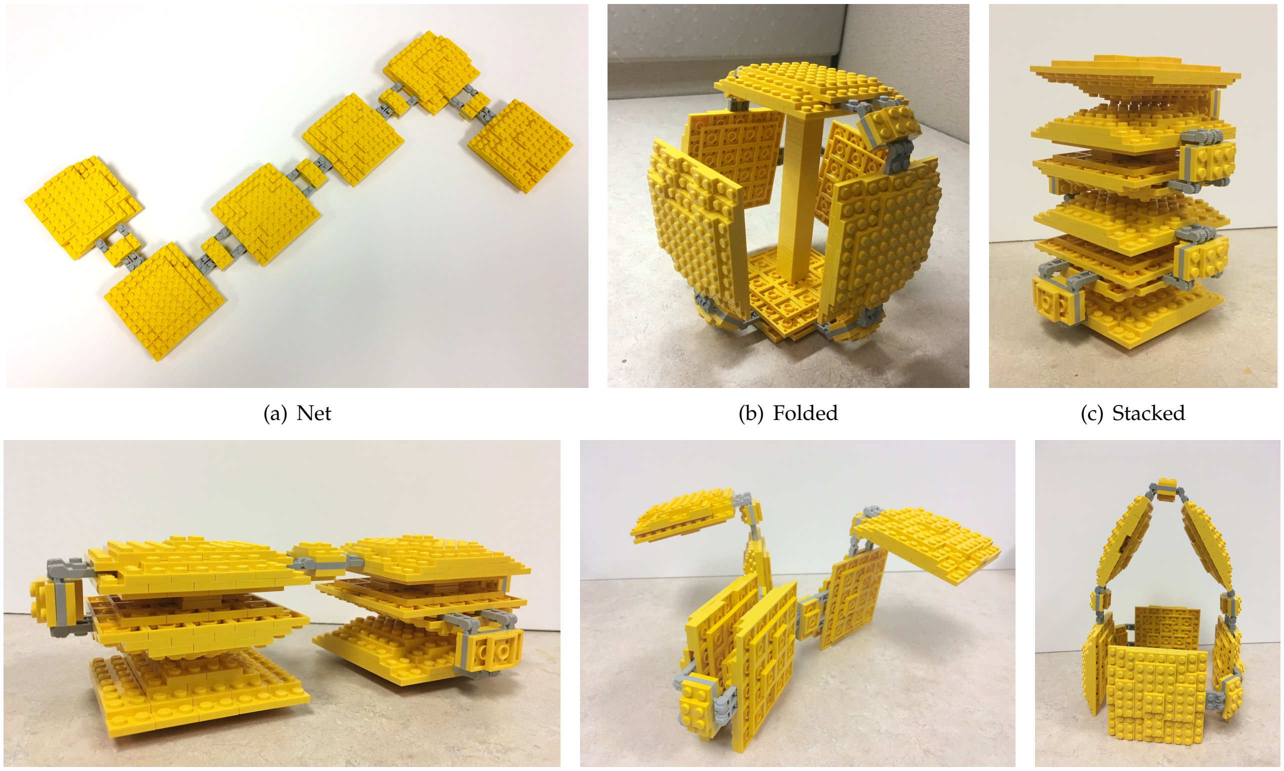
[13] LL Swanstrom, M Whiteford, and Y Khajanchee. Developing essential tools to enable transgastric surgery. *Surgical endoscopy*, 22(3):600–604, 2008.

[14] Tomohiro Tachi. Simulation of rigid origami. *Origami*, 4:175–187, 2009.

[15] Tomohiro Tachi. Rigid-foldable thick origami. 2011.

[16] Shigeo Takahashi, Hsiang-Yun Wu, Seow Hui Saw, Chun-Cheng Lin, and Hsu-Chun Yen. Optimized topological surgery for unfolding 3d meshes. In *Computer Graphics Forum*, volume 30, pages 2077–2086. Wiley Online Library, 2011.

[17] Gabriel Taubin. Constructing hamiltonian triangle



(a) Net

(b) Folded

(c) Stacked

(d) Stacked (2 piles)

(e) Plane

(f) House

Figure 12: A Lego realization of the cube model and two other shapes folded from the thick panel chain.

strips on quadrilateral meshes. In *Visualization and Mathematics III*, pages 69–91. Springer, 2003.

[18] Petr Vaněček and Ivana Kolingerová. Comparison of triangle strips algorithms. *Computers & Graphics*, 31(1):100–118, 2007.

[19] Zhonghua Xi, Yun hyeong Kim, Young J. Kim, and Jyh-Ming Lien. Learning to segment and unfold polyhedral mesh from failures. In *Shape Modeling International (SMI); also appears in Journal of Computers & Graphics*, Berlin, Germany, Jun. 2016.

[20] Zhonghua Xi and Jyh-Ming Lien. Continuous unfolding of polyhedra - a motion planning approach. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3249 – 3254, Hamburg, Germany, Sep. 2015.

[21] Yahan Zhou, Shinjiro Sueda, Wojciech Matusik, and Ariel Shamir. Boxelization: Folding 3d objects into boxes. *ACM Transactions on Graphics (TOG)*, 33(4):71, 2014.

[22] Shannon A Zirbel, Robert J Lang, Mark W Thomson, Deborah A Sigel, Phillip E Walkemeyer, Brian P Trease, Spencer P Magleby, and Larry L Howell. Accommodating thickness in origami-based deployable arrays. *Journal of Mechanical Design*, 135(11):111005, 2013.